ROBOTIC CAREGIVERS — SIMULATION AND CAPACITIVE SERVOING FOR PHYSICAL HUMAN-ROBOT INTERACTION

A Dissertation
Presented to
The Academic Faculty

By

Zackory Erickson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Robotics

School of Electrical and Computer Engineering
Georgia Institute of Technology

August 2021

ROBOTIC CAREGIVERS — SIMULATION AND CAPACITIVE SERVOING FOR PHYSICAL HUMAN-ROBOT INTERACTION

Thesis committee:

Prof. Charles C. Kemp, Advisor
Dept. of Biomedical Engineering
Georgia Institute of Technology

Prof. Greg Turk
School of Interactive Computing
Georgia Institute of Technology

Prof. C. Karen Liu
Computer Science Dept.
Stanford University

Prof. Pieter Abbeel
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley

Prof. Sonia Chernova
School of Interactive Computing
Georgia Institute of Technology

Date approved: July 6, 2021

Too often we underestimate the power of a touch, a smile, a kind word, a listening ear, an honest compliment, or the smallest act of caring, all of which have the potential to turn a life around.

Leo Buscaglia

To my family and lifelong partner

ACKNOWLEDGMENTS

I want to extend my greatest thanks and appreciation to my family who has always supported me and been my pillar throughout my undergraduate and graduate pursuits. This is especially true for my father Randy, my mother Korin, my sister Ashley, and my childhood friend, Beauty. I also wish to thank my partner, Hongting. It has been the joy of my life to grow with her through both of our PhDs. Her constant support and encouragement throughout the final years of my PhD have been remarkable. We have spent many late nights discussing research and her support was truly invaluable as I prepared for and ultimately accepted a faculty position. I couldn't have asked for a better life partner.

I am deeply grateful for my advisor, Prof. Charlie Kemp, who has led me through the wondrous world of robotics, healthcare, and graduate-level research. He has helped me through the roller coaster of graduate school. Charlie provided me with the freedom and support to continually explore new research directions, while simultaneously encouraging me to think critically about real-world problems. It has been so enjoyable writing grant proposals together, as well as developing and co-instructing a new course on Robotic Caregivers. He has been a source of inspiration and motivation, as a teacher, researcher, and mentor. I look forward to passing on what I have learned from Charlie to the next generations of students and researchers.

I am thankful to my committee members, Prof. Sonia Chernova, Prof. Karen Liu, Prof. Greg Turk, and Prof. Pieter Abbeel. I have thoroughly enjoyed working with each of them during my PhD, and I look forward to continuing these collaborations afterwards. I am also grateful for my funding support by the National Science Foundation (IIS-1514258, IIS-2024444), NIDILRR grant 90RE5016-01-00 via RERC TechSAge, as well as Amazon and Google for cloud credits that have enabled the computation for my research.

I have learned so much from my lab members and colleagues in the Healthcare Robotics Lab, including Tapomayukh Bhattacharjee, Henry Clever, Daehyung Park, Ariel Kapusta,

Phillip Grice, Yash Chitalia, and Patrick Grady. I would also like to extend a thank you to all of my friends and colleagues I have met throughout Georgia Tech and the RoboGrads organization. However, the list would be far too long to mention by name. A warm thank you to my roommate Matthew King-Smith. We have shared so many laughs and long discussions about research.

Finally, I have had the honor of mentoring an amazing group of undergraduate and Master's students in research. This includes Maggie Collier, Esther (Yijun) Gu, Pratyusha Karnati, Eliot Xing, Vamsee Gangaram, Bharat Srirangam, Nathan Luskey, Julia Chen, Katelyn Sosnowski, Mallak Taleb, Siyan (Sylvia) Li, Austin Jang, and Holden Schaffer. Several of these students are now doctoral students themselves, or on their way to soon becoming so.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Physical human-robot interaction and robotic assistance presents an opportunity to benefit the lives of many people, including the millions of older adults and people with physical disabilities, who have difficulty performing activities of daily living (ADLs) on their own. Robotic caregiving for activities of daily living could increase the independence of people with disabilities, improve quality of life, and help address global societal issues, such as aging populations, high healthcare costs, and shortages of healthcare workers. Yet, robotic assistance presents several challenges, including risks associated with physical human-robot interaction, difficulty sensing the human body, and complexities of modeling deformable materials (e.g. clothes).

We address these challenges through techniques that span the intersection of machine learning, physics simulation, sensing, and physical human-robot interaction.

Haptic Perspective-taking: We first demonstrate that by enabling a robot to predict how its future actions will physically affect a person (haptic perspective-taking), robots can provide safer assistance, especially within the context of robot-assisted dressing and manipulating deformable clothes. We train a recurrent model consisting of both a temporal estimator and predictor that allows a robot to predict the forces a garment is applying onto a person using haptic measurements from the robot's end effector. By combining this predictor with model predictive control (MPC), we observe emergent behaviors that result in the robot navigating a garment up a person's entire arm.

Capacitive Sensing for Tracking Human Pose: Towards the goal of robots performing robust and intelligent physical interactions with people, it is crucial that robots are able to accurately sense the human body, follow trajectories around the body, and track human motion. We have introduced a capacitive servoing control scheme that allows a robot to sense and navigate around human limbs during close physical interactions. Capacitive servoing leverages temporal measurements from a capacitive sensor mounted on a robot's end

effector to estimate the relative pose of a nearby human limb. Capacitive servoing then uses these human pose estimates within a feedback control loop in order to maneuver the robot's end effector around the surface of a human limb. Through studies with human participants, we have demonstrated that these sensors can enable a robot to track human motion in real time while providing assistance with dressing and bathing. We have also shown how these sensors can benefit a robot providing dressing assistance to real people with physical disabilities.

Physics Simulation for Assistive Robotics: While robotic caregivers may present an opportunity to improve the quality of life for people who require daily assistance, conducting this type of research presents several challenges, including high costs, slow data collection, and risks of physical interaction between people and robots. We have recently introduced Assistive Gym, the first open source physics-based simulation framework for modeling physical human-robot interaction and robotic assistance. We demonstrate how physics simulation can open up entirely new research directions and opportunities within physical human-robot interaction. This includes training versatile assistive robots, developing control algorithms towards common sense reasoning, constructing baselines and benchmarks for robotic caregiving, and investigating generalization of physical human-robot interaction from a number of angles, including human motion, preferences, and variation in human body shape and impairments. Finally, we show how virtual reality (VR) can help bridge the reality gap by bringing real people into physics simulation to interact with and receive assistance from virtual robotic caregivers.

CHAPTER 1
WHAT DOES THE PERSON FEEL? LEARNING TO INFER APPLIED FORCES DURING ROBOT-ASSISTED DRESSING

1.1 Research Summary

During robot-assisted dressing, a robot manipulates a garment in contact with a person's body. Inferring the forces applied to the person's body by the garment might enable a robot to provide more effective assistance and give the robot insight into what the person feels. However, complex mechanics govern the relationship between the robot's end effector and these forces. Using a physics-based simulation and data-driven methods, we demonstrate the feasibility of inferring forces across a person's body using only end effector measurements. Specifically, we present a long short-term memory (LSTM) network that at each time step takes a 9-dimensional input vector of force, torque, and velocity measurements from the robot's end effector and outputs a force map consisting of hundreds of inferred force magnitudes across the person's body. We trained and evaluated LSTMs on two tasks: pulling a hospital gown onto an arm and pulling shorts onto a leg. For both tasks, the LSTMs produced force maps that were similar to ground truth when visualized as heat maps across the limbs. We also evaluated their performance in terms of root-mean-square error. Their performance degraded when the end effector velocity was increased outside the training range, but generalized well to limb rotations. Overall, our results suggest that robots could learn to infer the forces people feel during robot-assisted dressing, although the extent to which this will generalize to the real world remains an open question.

1.2 Introduction

Survey data suggests that at least one million people in the United States require daily assistance with activities such as feeding, bathing and dressing [1]. Robotic assistance for these and other activities of daily living (ADLs) offers the potential for improved quality of life by improving independence, privacy and efficiency. In this chapter we focus on robot-assisted dressing, where a robot manipulates a garment in contact with a person receiving assistance.

While clothing often appears innocuous, garments can apply high forces to the human body and cause discomfort when pulled taut. The forces applied to the human body are also indicative of key aspects of the dressing task, such as which parts of the body are covered by the garment and whether or not the garment is caught on a part of the body. Directly measuring these forces through wearable sensors or instrumented clothing would be prohibitive. Since dressing is often intended to visually occlude parts of the body, visually estimating these forces would also be challenging. In this chapter, we investigate the possibility of a robot inferring these forces using only measurements from its end effector (see Figure 1.1). Specifically, our approach uses a 9-dimensional measurement vector, consisting of the 3-dimensional forces and torques applied to the robot's end effector by the cloth and the end effector's 3-dimensional Cartesian velocity.

Due to the complex mechanics relating the robot's end effector to the forces applied by the garment, we use a data-driven approach. We use a physics-based simulation to synthesize thousands of dressing examples during which we record measurements at the robot's end effector and the forces applied to the human body. We then use these data to train task-specific recurrent neural networks (RNNs) that infer the magnitudes of applied forces on the human body given end effector measurements over time. For example, at each time step, one of our networks estimates a 600-dimensional force map based on a 9-dimensional measurement vector. In this chapter we demonstrate that, in spite of the

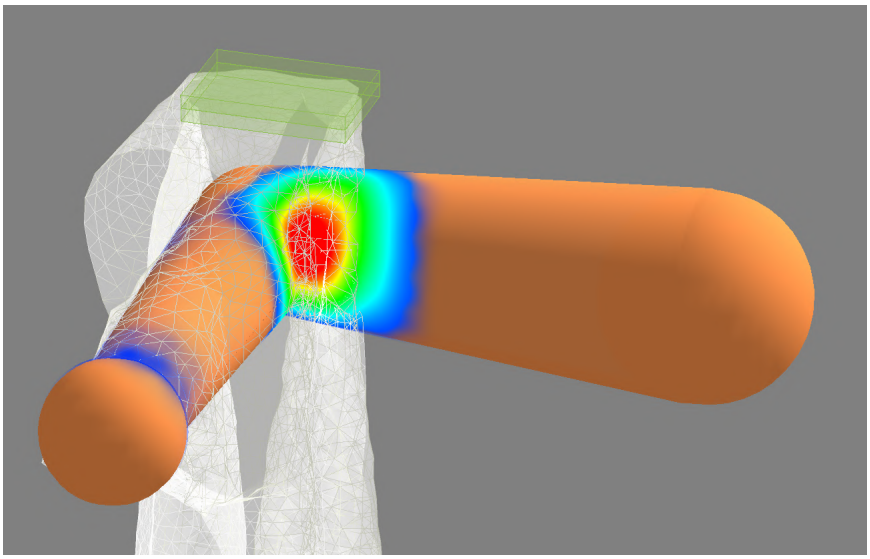


Figure 1.1: A force map estimated during a robot-assisted gown dressing task, using only the force, torque, and velocity measured at the end effector (shown in green). The semi-transparent cloth gown allows the force map underneath to be viewed.

low-dimensional input, high-dimensional output, and complex interactions involved, this inference can be performed in real time with compelling results which bodes well for future applications[1].

1.3 Related Work

Our approach is inspired in part by the notion that a human can use a lifetime of experience performing a task for him or herself to infer what someone else is feeling when assisted. This is related to work on mirror neurons [2]. The physics-simulation gives robots the opportunity to observe and learn forces applied to the human body during dressing.

A number of researchers have considered the problem of robot-assisted dressing. Often, the resulting systems have relied on visual data to make inferences about the state of cloth and dressing interaction. Kogatani et al. have proposed techniques for estimating human cloth topological relationships and cloth dynamics from motion capture and RGB-D data [3, 4, 5]. Twardon and Ritter considered RGB-D based cloth boundary detection for

[1] All code can be found at: https://github.com/gt-rad/learning-forces

garment grasping [6].

Others have used vision systems for robot-assisted dressing control tasks. Tamei et al. used motion capture to extract a low-dimensional cloth state representation based on the topological relationship between a mannequin and the garment. They then computed the arm motion for a dual arm robot to pull a T-shirt over the mannequin's head using reinforcement learning [7]. Klee et al. proposed a goal pose based system for a robot to assist users with dressing tasks. Using a turn taking strategy guided by a vision module, the robot and a user take turns improving their relative positions until a robot goal pose becomes feasible. They demonstrated this system with a Baxter robot assisting a user in donning a hat [8]. Gao et al. propose an approach for which Gaussian mixture models captured from RGB-D data approximate the movement space of a user's upper body joints. They then calculated poses for a robot to hold a garment within reach of a user to facilitate dressing of a sleeveless jacket [9].

While the use of vision as the primary form of input data has been common, the use of force data, if present, has been limited. Yamazaki et al. described an approach for assisted dressing failure detection through optical flow and force data [10]. Recently, Gao et al. have proposed a stochastic path optimization approach for personalized robot-assisted dressing that leverages both force and visual information [11]. Kapusta et al. have also shown that haptic data from a robot's end effector can be used to make predictions about the future state of a dressing task [12].

We use long short-term memory (LSTM) networks to perform estimations with time series data. LSTMs were first introduced by Hochreiter and Schmidhuber, and have since seen several improvements [13]. For instance, we use the forget gate proposed by Gers et al., which allows the network to learn to reset memory over time [14]. LSTMs have been used successfully throughout many applications, including: machine translation, generating cursive writing, and speech recognition [15, 16, 17].

Aviles et al. used dimensionality reduction and an LSTM to visually estimate force

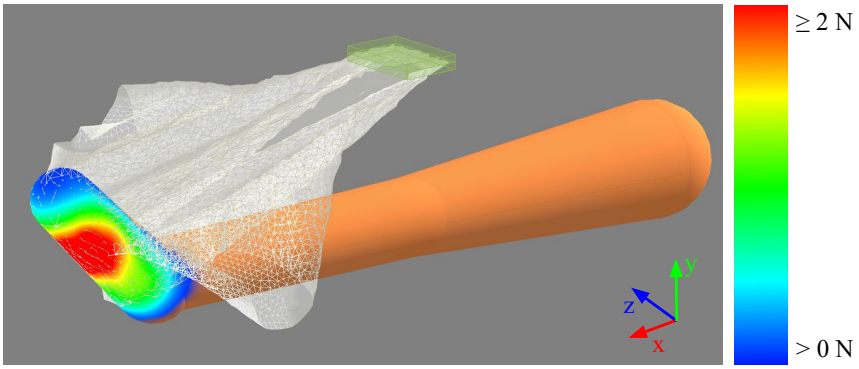


Figure 1.2: A full force map estimated along a human leg during a robot-assisted shorts dressing task.

Table 1.1: Measured and estimated forces during simulation at time t.

$f_{t,i}$	Magnitude of true force applied at limb point i
$\boldsymbol{f}^c_{t,j}$	The j^{th} collision force between the cloth and the limb
$\boldsymbol{f}^r_t$	3-dimensional force applied at the robot's end effector
$\hat{f}^u_{t,i}$	Unprocessed force mag. at limb point i, estimated by LSTM

applied by a robotic surgical tool to tissue over time. Their algorithm uses stereo images of the tissue and the motion of the surgical tool as input and outputs a single scalar force at each time step [18]. Berenson proposed an iterative Jacobian-based method to manipulate deformable objects without an explicit object model [19]. Similar to Berenson's work, our LSTM-based approach does not have access to a model of the deformable garment, yet is able to learn how this cloth interacts with a human body.

1.4 Force Map Estimation

In this work, we focus on the problem of estimating the magnitudes of the forces applied to a human limb by a cloth garment during a robot-assisted dressing task (see Figure 1.2). We refer to several different forces which we differentiate using superscripts as shown in Table 1.1.

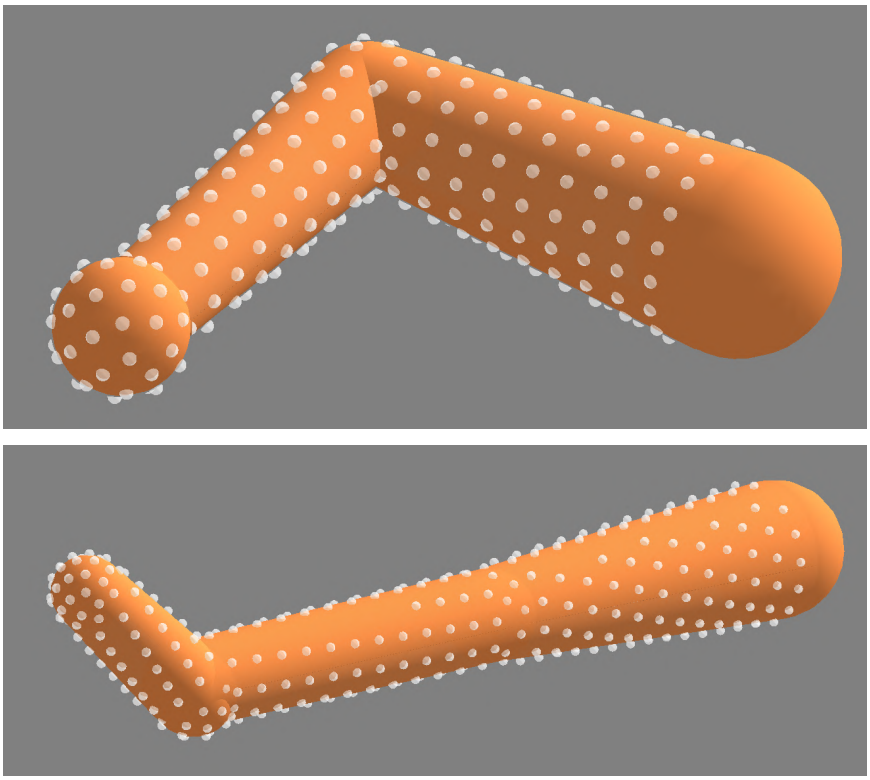


Figure 1.3: A force map consists of force magnitudes at discrete points along each limb which are shown here in white. We space these limb points approximately 2cm apart from each other with 300 points along the arm and 400 along the leg. For every time step, the LSTM estimates the magnitude of force applied at each of these points.

1.4.1 Problem Description

We define a force map as an N-dimensional vector of force magnitudes at fixed locations along a simulated limb. We use a set of uniformly distributed points, which we refer to as limb points, to represent these fixed locations (Figure 1.3).

We denote this force map at a time step T by $\boldsymbol{f}_T = \left[f_{T,1}, f_{T,2}, \cdots, f_{T,N}\right]$ where $f_{T,i}$ is the magnitude of force at the i^{th} limb point. We expect this force map to be densely sampled across the limb such that $\boldsymbol{f}_T$ has several hundred dimensions. At time T, there can be a set of collisions between the cloth and the limb where the j^{th} collision consists of a vertex along the cloth mesh with position $\boldsymbol{x}_{T,j}$ that applies a force, $\boldsymbol{f}^c_{T,j}$, onto the limb (Figure 1.4). Since these collisions may occur anywhere on the limb, we map the force magnitude of collisions to nearby limb points. We provide a more thorough formulation of

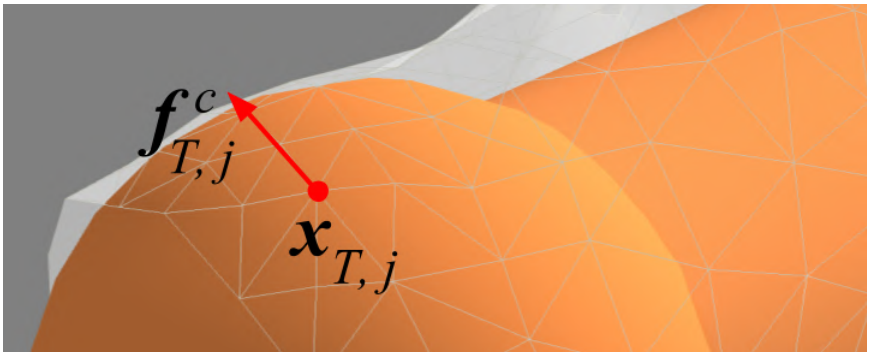


Figure 1.4: Visualization of a single point of contact between the cloth gown and the arm. While the cloth is pulled along the arm at time T, the j^{th} collision point has a global position, $\boldsymbol{x}_{T,j}$, and exerts a force, $\boldsymbol{f}^c_{T,j}$, onto the arm.

this mapping in Sec. 1.5.2.

In this work, we aim to learn a function that estimates $\boldsymbol{f}_T$ given a sequence of measurements seen thus far. At each time step $t \in \{1, \ldots, T\}$ we use a 9-dimensional measurement vector, $\boldsymbol{\phi}_t$, consisting of the 3-dimensional force, $\boldsymbol{f}^r$, and torque, $\boldsymbol{\tau}$, measured at the robot's end effector along with the 3-dimensional Cartesian velocity, $\boldsymbol{v}$, of the robot's end effector, i.e. $\boldsymbol{\phi}_t = [\boldsymbol{f}^r_t, \boldsymbol{\tau}_t, \boldsymbol{v}_t]$. Thus, we define the estimation problem as finding a function h that accurately estimates $\boldsymbol{f}_T$ given $\boldsymbol{\phi}_{1:T}$, i.e. $\boldsymbol{f}_T \approx h(\boldsymbol{\phi}_{1:T})$.

1.4.2 LSTM

We use LSTMs to perform this estimation problem, and hence serve the role of the function h. An LSTM is a type of recurrent neural network (RNN) [13]. We use an LSTM structure similar to that presented by Gers et al. in which the activation of memory cells are defined using three gates: an input gate, a forget gate, and an output gate [14].

One issue that arose as we designed LSTM networks for this estimation problem is a tendency for the LSTMs to estimate a very low, yet nonzero, force for limb points that were not in contact with the garment. This made it challenging to determine where contact was actually occurring along the limb and interfered with visualization of the force maps for evaluation. One approach to account for this issue would be to implement an arbitrary threshold below which all estimated force magnitudes would be set to 0. However, selecting

a threshold is not straightforward.

Instead, at each time step t, our LSTM estimates both an unprocessed force map, $\hat{\boldsymbol{f}}_t^u$, and a contact map, $\hat{\boldsymbol{\chi}}_t$. The contact map is used to make a binary classification of whether or not contact is occurring at each limb point. When creating our training set, for each ground truth force, $f_{t,j}$, at limb point j, the accompanying contact map element, $\chi_{t,j}$, within the training set is set such that,

$$\chi_{t,j} = \begin{cases} 0.1 & \text{if } f_{t,j} > 0, \\ -0.1 & \text{otherwise.} \end{cases} \tag{1.1}$$

In order to avoid bias between learning the contact map versus the force map, we weight the contact map values so they are near the values we expect within the force map (i.e. ~ 0.1). If we set the values within the contact map too large, then the LSTM learns to minimize the large errors within the contact map, but fails to minimize errors that occur in the force map. When the LSTM estimates a contact map, $\hat{\boldsymbol{\chi}}_t$, at time t, we can use this contact map to estimate which limb points are actually experiencing contact with the cloth (i.e. the j^{th} limb point is in contact with the cloth when $\hat{\chi}_{t,j} > 0$).

To estimate the true force map at a time step T, we provide the LSTM with a set of measurements, $\boldsymbol{\phi}_{1:T}$, for all previous time steps. Given $\boldsymbol{\phi}_{1:T}$, the trained LSTM then estimates an unprocessed force map and contact map, $\left[\hat{\boldsymbol{f}}_t^u, \hat{\boldsymbol{\chi}}_t\right]$, for each time step $t = 1, \ldots, T$. When visualizing real time force map estimations, we only consider the estimation for the most recent time step, $\left[\hat{\boldsymbol{f}}_T^u, \hat{\boldsymbol{\chi}}_T\right]$. Finally, the estimated contact map provides intuition into which limb points are experiencing contact and hence we use this information to compute the estimated force map, $\hat{\boldsymbol{f}}_T$. We define the elements of this force map such that for each

limb point i,

$$\hat{f}_{T,i} = \begin{cases} 0 & \text{if } \hat{\chi}_{T,i} < 0 \text{ or } \hat{f}^{u}_{T,i} < 0, \\ \hat{f}^{u}_{T,i} & \text{otherwise.} \end{cases} \tag{1.2}$$

$\hat{\boldsymbol{f}}_T$ is then a vector that represents an estimated force magnitude at each limb point with force set to zero for limb points not in contact with the cloth ($\hat{\chi}_{T,i} < 0$) and force magnitudes that were estimated to be negative ($\hat{f}^{u}_{T,i} < 0$).

Our task-specific LSTMs consist of 3 layers, each with 50 cells. The LSTM layers use a *tanh* activation function and we initialize the LSTM's parameters using Glorot's uniform distribution [20]. The network's output layer is fully connected with linear activations. Altogether, the LSTM has 52,400 recurrent connections. It is worth noting that our approach does not rely on any parsers or encoders, nor does it require any dimensionality reduction since our LSTM directly estimates an entire force map at each time step.

1.5 Physics-Based Simulation

We examine two robot-assisted dressing scenarios: pulling a hospital gown onto an arm and pulling a pair of shorts onto a leg, which we refer to as the gown task and shorts task, respectively. Our simulator extends work by Yu et al. in which they designed a simulator to model real world data from a robot pulling a hospital gown onto human participant's arms [21]. Each simulation was built in OpenGL with NVIDIA PhysX[2] for physics and cloth simulation. PhysX is robust to large forces applied to the cloth and is based on position-based dynamics. We made several modifications to the base PhysX software, including: calculating collision forces, improving friction calculations to better match real world data, and implementing friction for cloth self-collisions to allow wrinkles and folds within the garments. A more thorough account of these modifications can be found in [21].

[2]NVIDIA PhysX: https://developer.nvidia.com/physx-sdk

Simulating the 6-dimensional force and torque measurements relies on accurately modeling complex cloth dynamics and collisions. Because of this, we optimized the simulator using CMA-ES according to data collected from a real robot-assisted dressing task with human participants, as described in Yu et al. [21, 22]. We model the human arm and leg as spheres connected by conical frustums which allow for fast collision detection. We dimensioned the human limbs according to a 50% male. The limbs are rigid bodies fixed in place throughout a trial. We selected limb poses such that the robot must apply a reasonable amount of force in order to successfully perform the dressing task. The garments are represented as triangle meshes based on measurements of a real hospital gown and pair of shorts.

1.5.1 Task Variation

We incorporated variability into the garment's initialization by applying a small uniform force to each vertex along the garment for a short duration of time prior to a trial. At each time step t, we collect the measurements, $\boldsymbol{\phi}_t$, along with all ground truth collision positions $\boldsymbol{x}_t$ and force magnitudes $\boldsymbol{f}_t^c$ that occur between the cloth and the simulated body.

When recording training data for the gown task, we added variations that we expect in a real world robot-assisted dressing task. For instance, the initial location of the human's limb relative to the robot's end effector could vary for each dressing attempt. Additionally, the robot's end effector may vary in velocity and follow a nonlinear path while providing dressing assistance. To this end, between sequences, we varied the human limb location, the velocity of the end effector's motion, and the path that the end effector follows, for both dressing tasks. To vary the human arm location with respect to the robot's end effector we centered the human arm at the origin and then randomly varied its position ± 5 cm along the x axis, ± 20 cm along the y axis, and ± 10 cm along the z axis (see Figure 1.5 for axes), which we denote by $(\pm 5\text{cm}, \pm 20\text{cm}, \pm 10\text{cm})$. We selected these position ranges such that the cloth and human limb would be guaranteed to make contact during the simulation.

We initialized the robot's end effector at a fixed location above the arm and had it follow a nonlinear randomized cubic Hermite spline path (as shown in Figure 1.5) with 7 to 10 control points that deviated from a linear path by $(\pm 3\text{cm}, \pm 2\text{cm}, \pm 5\text{cm})$. Moreover, while the end effector maintained a constant speed during a dressing sequence, we varied the speed from 10 cm/s to 15 cm/s between dressing sequences. It is worth noting that while the magnitude of the end effector's velocity remains constant over time, the three individual components of the velocity do change as the end effector progresses through the spline path. The shorts dressing task varies in a similar fashion with the leg fluctuating from the origin by $(\pm 5\text{cm}, \pm 7.5\text{cm}, \pm 5\text{cm})$, the spline path varying by $(\pm 3\text{cm}, \pm 2\text{cm}, \pm 5\text{cm})$ with 9 to 11 control points, and the end effector speed varying between 15 cm/s and 20 cm/s. Due to the 45° angle of the foot, the end effector's unvaried path drops down closer to the leg when passing over the foot, as seen in Figure 1.5.

1.5.2 Generating the Ground Truth Force Map

As described in Sec. 1.4.1, we define a force map as a collection of force magnitudes at uniformly distributed limb points. The collisions between the cloth and limb will not align perfectly with our set of limb points as these collisions can occur at any location along the limb. As such, our approach proportionally distributes the force magnitude of each collision to the k nearest limb points. At the current time step T, it identifies the k nearest limb points to the j^{th} collision location, $\boldsymbol{x}_{T,j}$, and distributes the magnitude of collision force $|\boldsymbol{f}^c_{T,j}|$ to them. Specifically, it adds $\frac{1}{k-1}\left(1 - \frac{d_l}{\sum_{p=1}^{k} d_p}\right)|\boldsymbol{f}^c_{T,j}|$ to the magnitude of the l^{th} nearest limb point, where d_l is the Euclidean distance of the l^{th} limb point from $\boldsymbol{x}_{T,j}$ and $\sum_{l=1}^{k} \frac{1}{k-1}\left(1 - \frac{d_l}{\sum_{p=1}^{k} d_p}\right) = 1$.

For this work, we used $k = 5$ as we found that mapping a collision force to the nearest five limb points created a smoother force map and improved the LSTMs' performance.

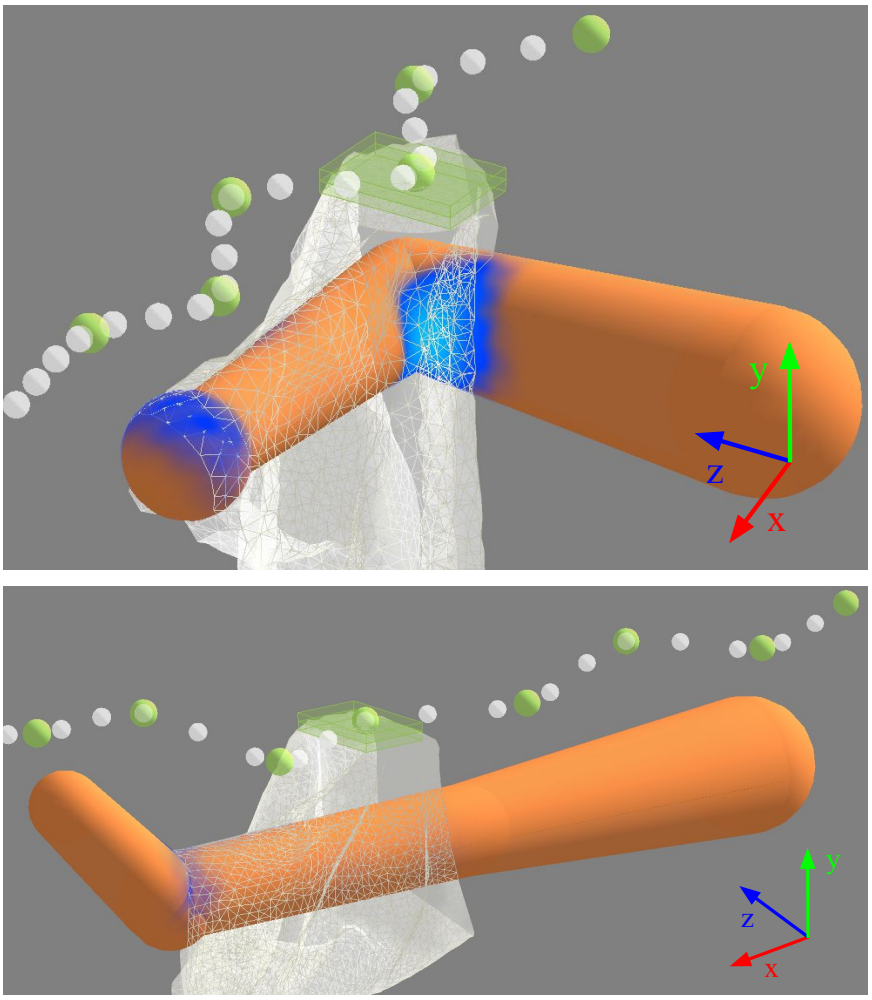


Figure 1.5: Example of a random spline trajectory followed by the end effector during both a gown and shorts dressing task. The green control points define the spline trajectory and white points show samples on the interpolated path.

1.6 Evaluation

For each task, we collected a training set consisting of 5000 dressing sequences and a test set of 1000 sequences. These sequences were randomized in limb position, end effector speed, and end effector path, as presented in Sec. 1.5.

We used Keras, a neural network library, to implement an LSTM for each dressing task [23]. We trained the LSTMs with stochastic gradient descent and adaptive learning rates using RMSprop and a mean squared error loss function [24]. We used batches of 16 sequences and trained for a total of 10 epochs. We performed all training on an Amazon

EC2 server with 36 cores and 64 GB of memory, taking approximately 2.5 hours to train each of the two LSTMs.

As defined in Sec. 1.4.2, the LSTM estimates both an unprocessed force map and corresponding contact map at each time step, which are used to produce the estimated force map. This amounts to estimating a 600-dimensional output vector for the gown task and an 800-dimensional vector for the shorts task at each time step.

Below, we describe how the LSTM performed on data which are similar to that collected for training and compare differences among the two tasks. We then analyze how our LSTM-based approach generalizes to unseen scenarios, such as higher end effector velocities, or rotations of the human limb. Finally, we present qualitative (visual) results of the estimated force maps in various scenarios, and present a few cases in which the estimated force maps deviate from the ground truth.

1.6.1 Visualizations

For both tasks, the LSTMs produced force maps that were noticeably similar to ground truth when visualized as heat maps across the limbs. Specifically, we visualized force maps by mapping each force magnitude to a constant color spectrum, as shown in Figure 1.2, and interpolating colors between points along the limb. Within this chapter, we present frames from a number of representative trials to help convey the performance of our approach. We also have generated videos that accompany this chapter.

The arm rarely gets caught in the gown sleeve as the arm either successfully enters the sleeve, or misses the sleeve for most dressing attempts. Contrary to this, the shorts frequently get caught on the foot during the shorts dressing task, which causes large forces as the end effector continues to pull.

Figure 1.6 shows a sequence of frames for a hospital gown dressing scenario in which the gown is successfully pulled up the forearm. Low forces are applied to the fist and forearm initially, while larger forces are applied to the elbow and upper arm as the task

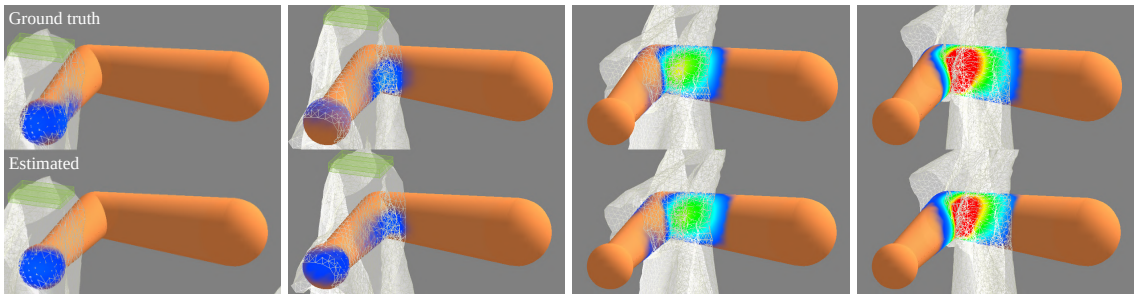


Figure 1.6: A sequence for the simulated gown task where the top arm represents the ground truth force map and the bottom arm depicts the LSTM estimated force map. The end effector used a fixed speed of 15 cm/s with an arm position offset of $(0\text{cm}, 20\text{cm}, 0\text{cm})$.

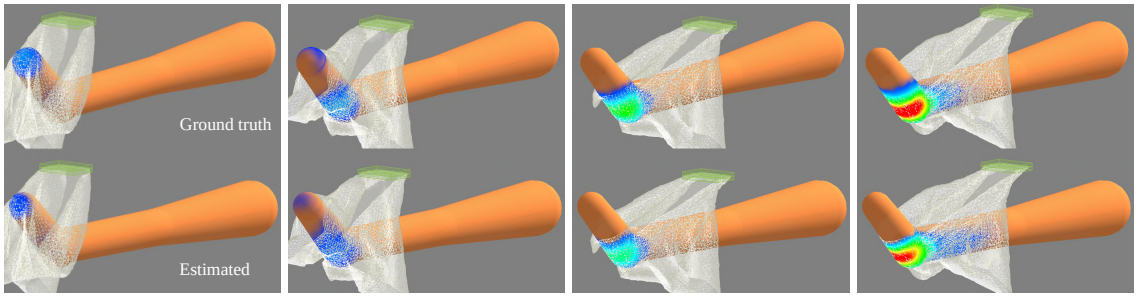


Figure 1.7: A sequence for the shorts task with the top and bottom legs showing the ground truth and estimated force maps respectively. The end effector's speed was 20 cm/s with a leg position offset of $(0\text{cm}, 4.5\text{cm}, 0\text{cm})$.

progresses. Figure 1.7 shows a similar sequence for the shorts dressing task. However, the shorts get caught on the heel of the leg early on, which causes large forces to be applied as the end effector continues to pull and stretch the cloth. Figure 1.8 visualizes a scenario in which the arm gets caught in the sleeve of the gown. Due to the large size of the gown, larger forces are still not applied until the latter end of the sequence. Figure 1.9 depicts a sequence in which the shorts are successfully pulled up the leg without getting caught. Figure 1.10 shows a sequence during which the arm misses the sleeve opening of the gown. Lastly, Figure 1.11 presents a sequence for which a higher speed of the robot's end effector led to errors in the estimated force maps.

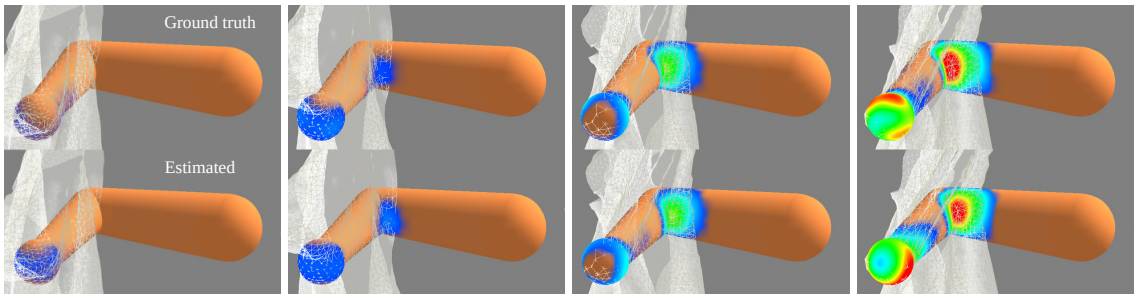


Figure 1.8: A visual sequence of the gown task for which the sleeve of the gown gets caught on the fist. The arm is positioned at an offset of $(0\text{cm}, 11\text{cm}, 0\text{cm})$ and the end effector has a speed of 20 cm/s which is greater than the range of speeds used to train the LSTM.

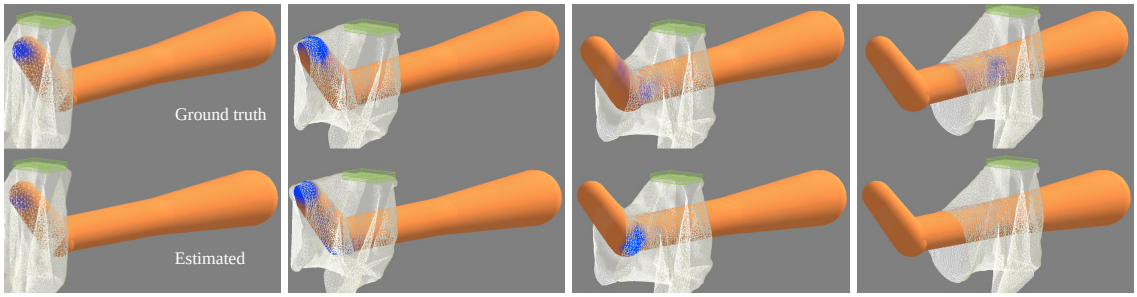


Figure 1.9: A shorts dressing sequence for which the shorts are successfully pulled over the foot and up the leg without getting caught. We use an end effector speed of 15 cm/s and a leg position offset of $(0\text{cm}, 7.5\text{cm}, 0\text{cm})$.

1.6.2 Root-Mean-Square Error (RMSE)

In addition to visualization, we computed a root-mean-square error (RMSE) metric between the estimated and ground truth force maps, averaged across every time step for all 1000 test sequences. We compute this RMSE as,

$$\text{RMSE} = \sqrt{\frac{1}{TN} \sum_t^T \sum_i^N (f_{t,i} - \hat{f}_{t,i})^2}$$

where T is the total number of time steps in a sequence, N is the number of limb points, and $f_{t,i}$ is the force for limb point i at time t. This metric provides the benefit of presenting results in units of Newtons and is motivated by the use of MSE as a loss function for the LSTM. Note that we only compute the RMSE for errors between the estimated force maps, $\hat{\boldsymbol{f}}_t$, and ground truth force maps, $\boldsymbol{f}_t$, and not between the contact maps. For the gown

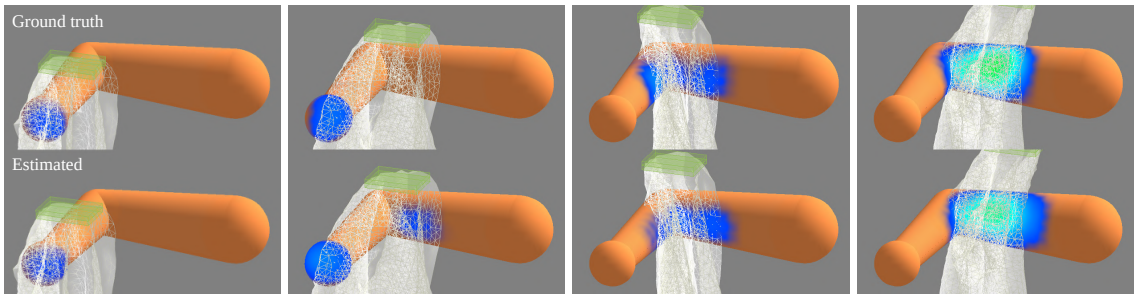


Figure 1.10: A simulated gown task sequence for which the arm misses the sleeve of the gown. We set the end effector speed to 15 cm/s and arm position offset to $(0\text{cm}, 20\text{cm}, 7.5\text{cm})$.

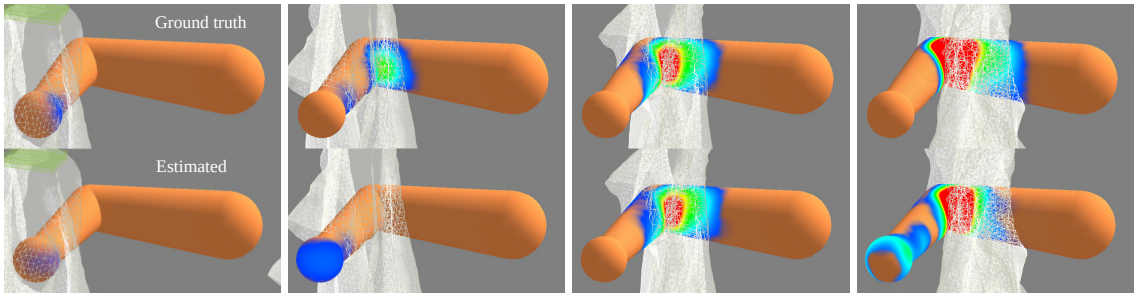


Figure 1.11: A gown task sequence which visualizes a scenario in which, for a few time steps, the estimated force maps differ from the ground truth force maps. We use an end effector speed of 20 cm/s which is outside the range of velocity magnitudes that the LSTM has been trained on. The arm is positioned at an offset of $(-5\text{cm}, 15\text{cm}, 0\text{cm})$.

dressing task, we computed an average RMSE value of 0.00673 N over all test sequences. We also computed an average RMSE value of 0.0488 N for the shorts dressing task. We use these values as a baseline for evaluating the LSTM's performance in generalizing to unseen scenarios in the following section. For comparison, both of these baseline values approximately double to 0.0148 N and 0.102 N, respectively, if we use a zero collision force map estimation for all test sequences, i.e. $\hat{f}_{t,i} = 0$.

1.6.3 Generalization to Unseen Scenarios

While the LSTM performs well with scenarios similar to the training set, we are also interested in how well our approach generalizes since real world dressing will vary in numerous ways. To this end, we compare the RMSE scores between estimated and ground truth force maps for faster end effector velocities and for when the forearm and foot are rotated. Note

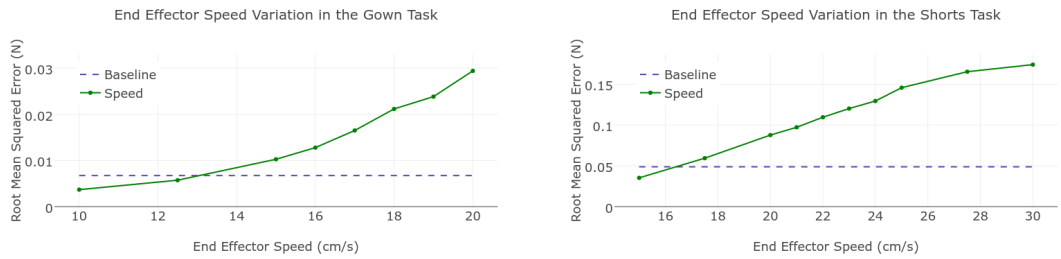


Figure 1.12: The figure shows the RMSE between the estimated force maps, $\hat{\boldsymbol{f}}_t$, and the ground truth force maps, $\boldsymbol{f}_t$, for several speeds beyond what the LSTM has seen. RMSE values are averaged over 128 sequences (N=128) with varied limb positions and spline trajectories.

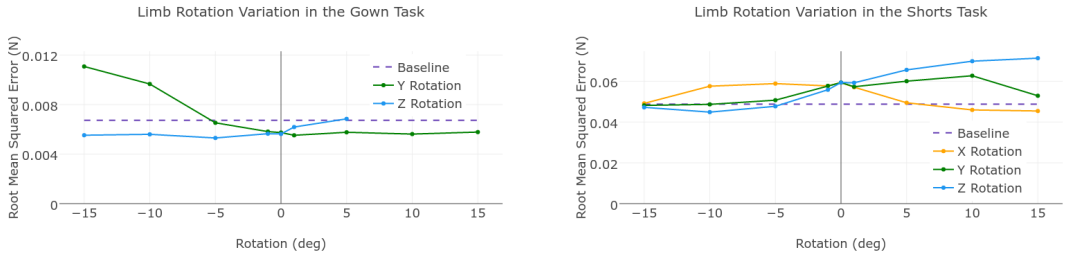


Figure 1.13: The RMSE between the estimated force maps, $\hat{\boldsymbol{f}}_t$, and the ground truth force maps, $\boldsymbol{f}_t$, for numerous rotations of the simulated forearm rotated around the elbow and foot rotated around the ankle. For the gown task, we limit rotations along the positive Z-axis to prevent collisions between the arm and end effector. Additionally, the forearm is parallel to the X-axis. As such, we do not rotate around this axis as it would not impact the arm's pose. RMSE values are averaged over 128 sequences (N=128) with varied limb positions and spline trajectories.

that the LSTM has never seen a limb rotation, hence the baseline is denoted by a rotation of $0°$.

Figure 1.12 highlights how the RMSE varies as we increase the end effector speed for both the gown and shorts tasks. To maintain consistency within each dressing task, we tested each end effector speed on the same 128 randomly generated sequences that varied in limb position and spline trajectory.

While training for the gown task, we used end effector speeds that ranged from 10 cm/s to 15 cm/s. With a speed of 20 cm/s, the RMSE of 0.0294 N is approximately 4.4 times larger than the baseline RMSE for the gown task. For comparison, the dressing sequence in Figure 1.6 has an RMSE value of 0.019 N. For the shorts dressing task, we trained the

LSTM on speeds ranging from 15 cm/s to 20 cm/s. At 25 cm/s, the RMSE is ~ 3 times larger than the baseline, and this rises to a factor of 3.6 with a speed of 30 cm/s.

Analogous to high end effector speeds, Figure 1.13 depicts this same metric for several forearm and foot rotations. Each rotation was tested on the same 128 randomly generated sequences with varied limb positions and spline trajectories. However, we used an end effector speed of 12.5 cm/s and 17.5 cm/s for the gown and shorts tasks respectively. When the forearm is rotated -15° around the Y-axis the RMSE increases to 1.65 times the gown task baseline, whereas a rotation of -15° around the Z-axis generates a RMSE of 0.82 times the baseline. Finally, all of the leg rotations across all 3 axes presented for the shorts task ranged between a factor of 0.92 and 1.46 times the shorts task baseline.

1.7 Discussion

In Figure 1.12, we observe large error increases as end effector speeds continue to extend beyond the range of speeds in the training set. This provides some indication that an LSTM-based approach may experience challenges estimating task progression when end effector speeds exceed those used for training. However, this increased error may also be due in part to higher contact forces from faster velocities.

Nevertheless, the LSTMs' RMSE performance generalized well to new limb rotations, despite not training on any rotations. In Figure 1.13, we observe a decrease in RMSE for some rotations. This could be influenced by some rotations reducing the magnitude of collision forces between the garment and limb. For example, the foot flattens with negative rotations along the Z-axis which lowers the chance of the shorts getting caught and exerting high forces on the foot.

We note that further work is necessary to evaluate how well real robots can use this approach to infer the forces a person feels during assistive tasks.

1.8 Conclusion

Inferring the forces applied to a person's body by a garment during robot-assisted dressing could give a robot insight into what a person physically feels and enable the robot to provide more effective assistance. In this work, we showed how task-specific LSTMs can estimate force magnitudes along a human limb for two simulated dressing tasks. At each time step our LSTM networks take a 9-dimensional input vector consisting of the force and torque applied to the end effector by the garment and the velocity of the end effector. The networks then output a force map at each time step consisting of hundreds of inferred force magnitudes across the person's body.

For both dressing tasks, our approach produced force maps that were visually similar to ground truth. We then explored how well this approach may generalize to unseen end effector velocities and limb rotations. A promising characteristic that emerged was the LSTM's ability to estimate force maps during scenarios in which the human limb was rotated. Notably, we have shown that high-dimensional force maps can be estimated using a sequence of low-dimensional measurements. These results suggest that robots could learn to infer the forces that people physically feel during robot-assisted dressing.

CHAPTER 2
DEEP HAPTIC MODEL PREDICTIVE CONTROL FOR ROBOT-ASSISTED DRESSING

2.1 Research Summary

Robot-assisted dressing offers an opportunity to benefit the lives of many people with disabilities, such as some older adults. However, robots currently lack common sense about the physical implications of their actions on people. The physical implications of dressing are complicated by non-rigid garments, which can result in a robot indirectly applying high forces to a person's body. We present a deep recurrent model that, when given a proposed action by the robot, predicts the forces a garment will apply to a person's body. We also show that a robot can provide better dressing assistance by using this model with model predictive control. The predictions made by our model only use haptic and kinematic observations from the robot's end effector, which are readily attainable. Collecting training data from real world physical human-robot interaction can be time consuming, costly, and put people at risk. Instead, we train our predictive model using data collected in an entirely self-supervised fashion from a physics-based simulation. We evaluated our approach with a PR2 robot that attempted to pull a hospital gown onto the arms of 10 human participants. With a 0.2s prediction horizon, our controller succeeded at high rates and lowered applied force while navigating the garment around a persons fist and elbow without getting caught. Shorter prediction horizons resulted in significantly reduced performance with the sleeve catching on the participants' fists and elbows, demonstrating the value of our model's predictions. These behaviors of mitigating catches emerged from our deep predictive model and the controller objective function, which primarily penalizes high forces.

2.2 Introduction

Robotic assistance presents an opportunity to benefit the lives of many people with disabilities, such as some older adults. However, robots currently lack common sense about the physical implications of their actions on people when providing assistance. Assistance with dressing can improve a person's quality of life by increasing his or her independence and privacy. Yet, dressing presents further difficulties for robots due to the complexities that arise when manipulating fabric garments around people.

Model predictive control (MPC) enables robots to account for errors and replan actions in real time when interacting in dynamic environments. For example, MPC has found success in several contexts such as obstacle avoidance and object manipulation [25, 26, 27]. However, these existing robotic controllers do not take into consideration the physical implications of a robot's actions on a person during physical human-robot interaction. This is especially true during robot-assisted dressing in which a robot may never make direct physical contact with a person, but instead apply force onto the person through an intermediary non-rigid garment. Yet, robots could greatly benefit from predicting the physical implications of their actions when interacting with people.

In this chapter, we propose a Deep Haptic MPC approach that allows a robot to minimize the predicted force it applies to a person during robotic assistance that requires physical contact. We train a recurrent model that consists of both an estimator and predictor network in order to predict the forces applied onto a person, and we detail the benefits of this approach in Chapter 2.4. The estimator outputs the location and magnitude of forces applied to a person's body given haptic sensory observations from a robot's end effector. The predictor outputs future haptic observations given a proposed action. Together, these two networks allow a robot to determine the physical implications of its actions by predicting how future actions will exert forces onto a person's body. We demonstrate our approach on a real robotic system that assisted 10 human participants in pulling a hospital gown onto

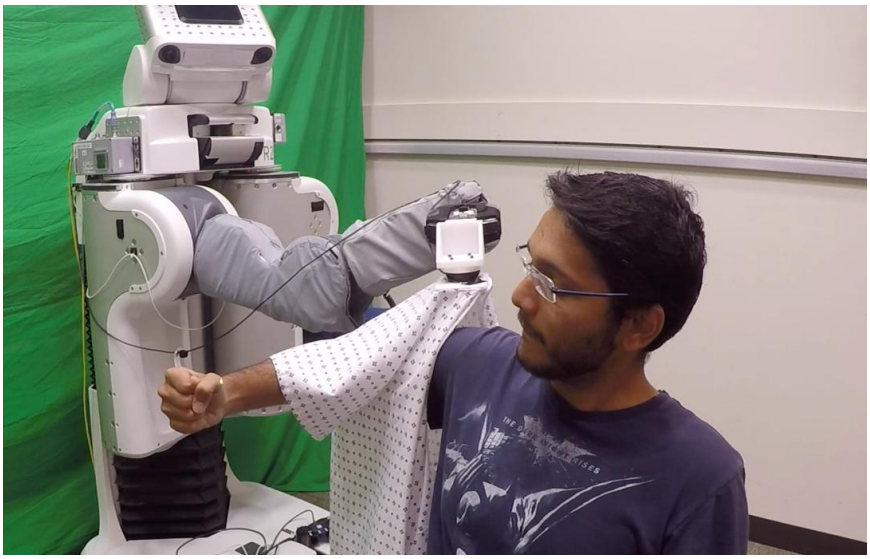


Figure 2.1: Using our approach, a PR2 pulls a hospital gown onto a participant's arm by minimizing the predicted forces applied to the person's body.

a person's right arm, as seen in Figure 2.1.

We train our model on data generated entirely in a physics-based simulation, allowing us to quickly collect thousands of diverse training sequences that would otherwise be dangerous or infeasible to collect on real robotic systems that physically interact with people. Our simulated robot can make mistakes, explore new approaches for interaction, and investigate error conditions without putting real people at risk.

These training data are generated in a self-supervised fashion, without a reward function or specified goal. Once training is complete, we define an objective function that enables our controller to prioritize future actions that minimize the predicted force applied to a person during dressing. Since our model is trained without a predefined reward function, we can redefine the objective function without retraining the model. We further compare dressing results for various time horizons with MPC and observe emergent behaviors as the prediction horizon increases.

The key contribution of this chapter is to demonstrate that a deep recurrent model over haptic and kinematic measurements can be used by real robotic systems to predict the physical implications of future actions and lower the forces applied to a person during

robot-assisted dressing. We show that this model can be trained in simulation and applied to a real robotic task of pulling a garment onto a person's arm. By combining our learning-based model with MPC, we observe emergent behaviors that result in the robot navigating a garment up a person's entire arm.

2.3 Related Work

2.3.1 Robot-Assisted Dressing and Force Estimation

Several robotic dressing approaches have relied on visual systems to estimate a person's pose and the state of a garment. For example, Koganti et al. [28] used RGB-D and motion capture data to estimate the topological relationship between a person's body and a garment. Klee et al. [8] visually detected a person's pose which was used by a Baxter robot to assist in putting on a hat. Pignat et al. [29] tracked a person's hand movement in real time using an AR tag. The researchers then used a Baxter robot to pull one sleeve of a jacket onto a person's arm. Unlike this body of work, our approach does not rely on visual observations, but is instead able to fully dress a person's arm using only haptic and kinematic measurements obtained at the robot's end effector.

Several researchers have similarly explored haptic sensing within the context of robot-assisted dressing. Gao et al. [11] proposed a force feedback control approach that allowed a Baxter robot to assist in dressing a sleeveless jacket. Kapusta et al. [12] explored how haptic observations at a robot's end effector can be coupled with an HMM to predict the future outcome of a dressing task. Yamazaki et al. [10] described a failure detection approach for robot-assisted dressing that leveraged force data while assisting participants in pulling up pants. Instead, our work demonstrates that haptic sensing and learning can be used to predict the physical implications of a robot's future actions when assisting people. When coupled with MPC, we show that these predictions also enable a robot to replan its actions in real time during robotic assistance.

In prior work [30] we presented an LSTM model trained in simulation to estimate the

forces applied onto a simulated arm and leg during robot-assisted dressing tasks. The estimator we present in this chapter uses a similar network architecture except we also provide end effector position and yaw rotation measurements to the model so that our PR2 can navigate around a person's elbow. We pair this estimator with a predictor and evaluate a PR2's ability to predict the physical implications of its actions during assistance.

2.3.2 Model Predictive Control

Model predictive control has found success in several robotics domains. Some examples include aerial control vehicles [31, 32] and robot locomotion [33, 34]. This work has similarity to [35, 26, 36], using haptic information as a model input for control in the manipulation domain. Prior robotics research has used analytical models for MPC [35, 37, 36, 26], whereas we employ a learning-based model as in [38, 39, 27].

Many past works have relied on vision-based approaches for robotic control with MPC. Finn and Levine [25] combined a predictive model of image observations with MPC for nonprehensile pushing tasks. Watter et al. [40] presented a learning-based control method for non-linear dynamical systems using raw pixel images. Boots et al. [41] learned a predictive model that generates RGB-D images of a robot arm moving in free space. In comparison to these vision-based methods, our learned model uses only haptic and kinematic information. Chow et al. [35] leveraged haptic observations with MPC to assist in reposition a person's limbs in simulation. Lenz et al. [27] learned material properties for cutting various foods with a PR2, but rely on joint torques for haptic feedback, which have a lower dimensionality and accuracy than the 6-DoF discrete force/torque sensor in our system. In addition, Jain et al. [26] showed how a robot arm can reach into cluttered spaces using haptic sensing skin.

Learning models with neural networks for robot control is common throughout many robotic control approaches [39]. Lenz et al. [27] used a recurrent model with MPC and demonstrated their approach on a PR2 that learned deep latent material properties by per-

forming 1,488 cuts across 20 foods. Finn and Levine [25] combined a deep predictive model of image sequences with MPC and trained their model on 50,000 pushing attempts of objects using 7-DoF manipulators. Unlike these approaches, our model is trained entirely in simulation, which presents several benefits for physical human-robot interaction, as discussed in Chapter 2.4.

Fu et al. [38] used model-based reinforcement learning in simulation, where a PR2 learned to manipulate rigid objects with MPC. Unlike reinforcement learning, our method does not require a reward function during training, which allows us to decouple the objective function from the learned model. Furthermore, we show that our learning-based model can enable a real PR2 to predict the physical implications of its actions when assisting human participants with dressing.

2.4 Simulation and Model Training

To perform deep haptic MPC that considers forces applied to a person, our model consists of two recurrent neural networks trained on a dataset of simulated robot-assisted dressing trials. Here we introduce notation and we provide a brief description of the model, simulation, and data collection process. Our dataset consists of 10,800 dressing trials generated in a simulated robot-assisted dressing environment presented in prior work [30]. As shown in Figure 2.2, this physics-based simulation consists of a robotic end effector that pulls a hospital gown onto a simulated human arm. The colored fields along the arm represent a force map that encompasses a set of force magnitudes applied at specific locations on the body. Several advantages arise from collecting data with a physics-based simulation. First, we can easily parallelize data collection to collect thousands of dressing experiences in a few hours. We can also test anomalous scenarios that may be infeasible or dangerous to test with real people, such as cloth getting caught on a body part. These anomalous conditions could be especially valuable for a robot, so that it can learn to mitigate potentially harmful consequences. Finally, we can calculate the location and magnitude of all forces applied to

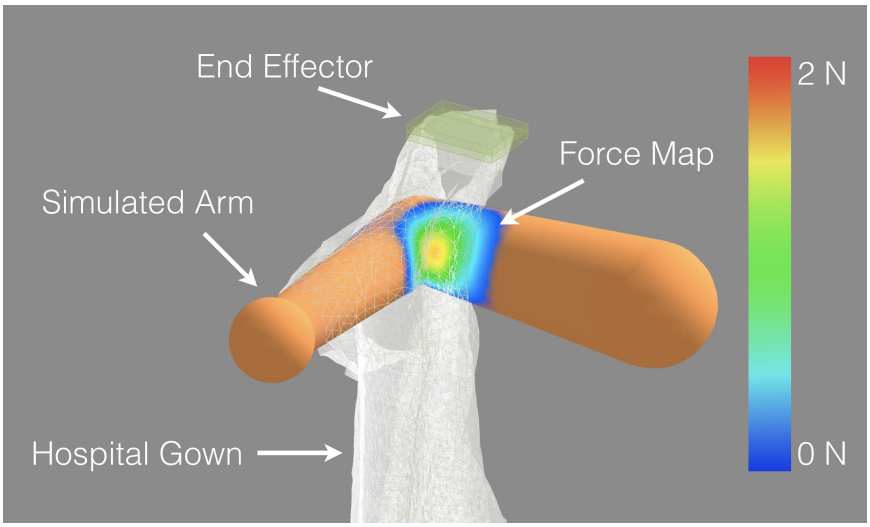


Figure 2.2: The physics-based simulation environment used to generate a training set for our model. The simulation records force, torque, position, and velocity from the robot's end effector, as well as all forces applied onto the arm by the fabric mesh gown.

a person by a clothing garment within simulation, something that is highly challenging in the real world.

During data collection, the simulated robotic end effector attempts to pull the sleeve of a hospital gown onto the person's arm. The simulator randomly selects a starting position near the arm and movement velocity for the end effector prior to each trial. During a dressing trial, the simulation iteratively selects a new random action for the robot's end effector at each time step. In doing so, our model learns about diverse situations for a garment to make contact with a person's arm. We represent actions in the fixed coordinate frame of the robot's torso, and actions consist of a 3D velocity for the end effector and a change in yaw rotation around an axis parallel to gravity, i.e. $\boldsymbol{a} = (v_x, v_y, v_z, \Delta\psi)$. The simulation selects new actions at 5 Hz and records sensor measurements at 100 Hz. Measurements $\boldsymbol{x}_t = (\boldsymbol{\rho}, \boldsymbol{v}, \psi, \boldsymbol{f}^r, \boldsymbol{\tau}^r) \in \mathbb{R}^{13}$, at time t, include the 3D position $\boldsymbol{\rho}$, 3D velocity $\boldsymbol{v}$, and yaw rotation ψ of the end effector, and the 3D forces $\boldsymbol{f}^r$ and torques $\boldsymbol{\tau}^r$ applied at the robot's end effector by the garment. We record all forces applied to the human's arm, which occur when a vertex on the fabric mesh makes contact with the simulated arm. We construct a force map, as shown in Figure 2.2, by mapping these applied forces to a discrete set of

fixed points (taxels) spaced across the surface of the arm. [30] provides further details of this mapping procedure and force map definition. In this work, we use 37 taxels distributed across the fist, forearm, and upper arm.

As Yu et al. [21] proposed, we used Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [22] to optimize the parameters of our simulator with respect to data collected from a real robotic system that assisted human participants in pulling on a hospital gown. Some of these parameters include garment stretch, stiffness, shear forces, and friction. Because of this optimization, the force and torque measurements in simulation align closely to those observed in the real world. However, the simulated end effector performs exact movements, whereas the motion trajectory of a PR2's end effector often includes noise due to the compliant nature of the arms. To account for this, we added a small amount of uniformly sampled noise, $\xi \in [-0.8, 0.8]$ mm/s, to each component of the end effector's velocity at every time step in the simulation. During model training, this also serves as a form of regularization to help mitigate overfitting to the measurements from simulation.

We leverage a pair of recurrent networks to predict the forces applied to a person given a sequence of proposed robot actions. We define a predictor $G(\boldsymbol{x}_{1:t}, \boldsymbol{a}_{t+1:t+H_p})$, which predicts a sequence of future end effector haptic measurements, $\hat{\boldsymbol{x}}_{t+1:t+H_p}$, that result from the robot executing actions $\boldsymbol{a}_{t+1:t+H_p}$ over a prediction horizon H_p. We then use an estimator, $F(\boldsymbol{x}_{1:t}, \hat{\boldsymbol{x}}_{t+1:t+H_p})$, that estimates the forces, $\boldsymbol{f}_{t+H_p}$, applied to a person at time $t + H_p$ given all prior measurements $\boldsymbol{x}_{1:t}$, and the predicted measurements, $\hat{\boldsymbol{x}}_{t+1:t+H_p}$. We can predict future force maps by composing the estimator and predictor, $F \circ G = F(\boldsymbol{x}_{1:t}, G(\boldsymbol{x}_{1:t}, \boldsymbol{a}_{t+1:t+H_p}))$, wherein we estimate force maps given predicted haptic measurements. Furthermore, we can make predictions beyond time $t + H_p$ by feeding the predicted measurements $\hat{\boldsymbol{x}}_{t+1:t+H_p}$ back into G along with an action sequence $\boldsymbol{a}_{t+H_p+1:t+2H_p}$. Thus, $\hat{\boldsymbol{x}}_{t+H_p+1:t+2H_p}$ can be predicted via $G(\{\boldsymbol{x}_{1:t}, \hat{\boldsymbol{x}}_{t+1:t+H_p}\}, \boldsymbol{a}_{t+H_p+1:t+2H_p})$.

Although these two networks could be merged, there are several advantages to a split architecture. First, this setup allows for additional flexibility in that a new predictor can

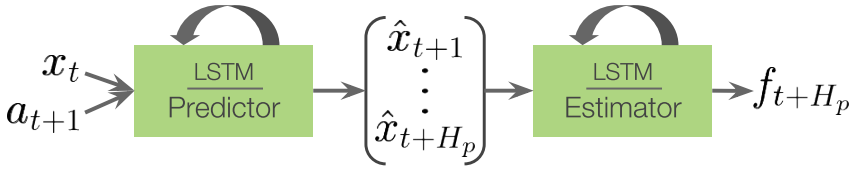


Figure 2.3: Our estimator and predictor LSTM networks with all associated inputs and outputs.

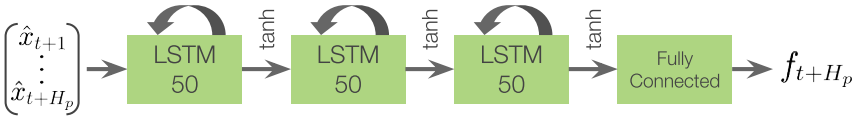


Figure 2.4: Network architecture for the estimator.

be learned without impacting the accuracy of force map estimation, or vise versa. Furthermore, we are able to run these two networks at different frequencies which is beneficial during real time use. We run the estimator at 100 Hz as this results in greater accuracy and resolution for force map estimates. However, since we update the robot's action at 5 Hz, we need only make predictions at a 5 Hz rate. In total, the estimation model receives 20 measurements for each step of the prediction model. This difference in frequencies was crucial for real-time implementation since prediction is a computationally demanding task for each candidate action.

This approach also presents several advantages over formulating a reinforcement learning problem and solving for a policy. Our objective function is decoupled from the learned model, thus the objective can be redefined for different dressing tasks without retraining the model. Also, the data we collect for training the estimator can be reused to train the predictor, whereas model-free reinforcement learning methods require further data collection and new sets of rollouts from the evolving policy after training an estimator.

As shown in Figure 2.3, our model uses LSTMs to estimate force maps and predict future measurements. Each recurrent model consists of three LSTM layers with 50 recurrent cells and a *tanh* activation. The final output layer is fully connected with a linear activation. Figure 2.4 shows this network architecture for our estimator model. The predictor

uses the same architecture, but with different input/output. Note that if the robot maintains a constant action throughout the entire prediction horizon, as is the case in our work, a sequence of identical actions, $\boldsymbol{a}_{t+1:t+H_p}$, can be collapsed down to a single action, $\boldsymbol{a}_{t+1}$. Because of this, our predictor outputs a sequence of measurements $\hat{\boldsymbol{x}}_{t+1:t+H_p}$ given a single action, $\boldsymbol{a}_{t+1}$, and measurement, $\boldsymbol{x}_t$. We use $H_p = 20$, which aligns with the 5 Hz rate used for predictions when the difference between time steps is 0.01s. Ideally, the predictor would evaluate sequences of actions that vary over time. However, our experiments showed that using the same action over the entire prediction horizon was computationally tractable and worked well in practice for physical human-robot interaction. Related literature has also found a 5 Hz action replanning rate to be computationally feasible for MPC on real robots [25].

2.5 Model Predictive Control

Our system uses model predictive control (MPC) with our recurrent estimator and predictor to choose actions that minimize the predicted force applied to a person during physical assistance. Here we present the cost function that we used to encourage certain robot actions and we describe ways in which this function could be adapted to allow for personalized robotic assistance. In addition, we present our MPC method for replanning actions, which involves predicting applied forces for a set of candidate robot actions.

We define a cost function leading to lower forces applied to a person during dressing assistance. The cost function input includes the current and prior measurements, $\boldsymbol{x}_{1:t}$ and a sequence of candidate actions, $\boldsymbol{a}_{t+1:t+H_p}$. In addition to penalizing large forces applied on the person's body, the cost function encourages forward moving end effector actions and

penalizes yaw rotations, represented by three weighted terms:

$$
\begin{aligned}
J(\boldsymbol{x}_{1:t}, \boldsymbol{a}_{t+1:t+H_p}) = w_1 \left\| F(\boldsymbol{x}_{1:t}, G(\boldsymbol{x}_{1:t}, \boldsymbol{a}_{t+1:t+H_p})) \right\|_1^2 \\
- w_2 \sum_{j=t+1}^{t+H_p} \bar{\boldsymbol{d}} \cdot \boldsymbol{a}_{j,v} \\
+ w_3 \sum_{j=t+1}^{t+H_p} |a_{j,\psi}|
\end{aligned}
\tag{2.1}
$$

where $\boldsymbol{a}_j = (v_x, v_y, v_z, \Delta\psi)$ represents a candidate action, $\boldsymbol{a}_{j,v}$ represents the 3-axis velocity components of the action, $a_{j,\Delta\psi}$ represents the yaw rotation component of the action, $\bar{\boldsymbol{d}} = (1, 0, 0)$ depicts a forward moving action, and w_1, w_2, w_3 are constant weights set based on the importance of making task progress versus keeping forces low. The first term, $\left\| F(\boldsymbol{x}_{1:t}, G(\boldsymbol{x}_{1:t}, \boldsymbol{a}_{t+1:t+H_p})) \right\|_1$, represents the L_1 norm of all predicted forces, $\boldsymbol{f}_{t+H_p}$, at the 37 taxels along a person's arm at time $t + H_p$. We square this term to reduce the influence of small forces that occur at the beginning of dressing. This is supported by the notion that small forces are unlikely to cause issues during assistance [26]. However, as more force is applied to a person's arm, this term becomes the dominating factor for selecting which action the robot will execute. For various applications, this L_1 norm term may also be modified to focus on certain body joints, e.g. only minimizing force around the hand and wrist, rather than the entire arm. The second term, $\bar{\boldsymbol{d}} \cdot \boldsymbol{a}_{j,v}$, rewards actions that move in a forward direction along the +X global coordinate axis, or approximately the central axis of a person's forearm, as shown in Figure 2.5. The last term, $|a_{j,\psi}|$, penalizes actions that perform a yaw rotation. Without these last two terms, the optimal action to minimize cost is sometimes an action that performs no movement. Depending on the task, the terms in Equation (2.1) may also be combined via a nonlinear function to support a variety of complex behaviors. From our experiments described in Chapter 2.6, we observe that this simple cost function can lead to emergent behaviors in which the robot can navigate a garment up

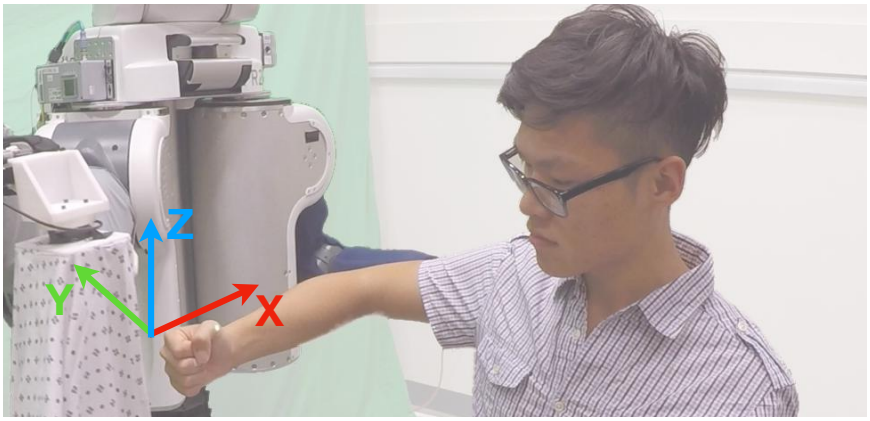


Figure 2.5: Initial configuration for the robot and a participant prior to the start of a dressing trial. We overlay the global coordinate axes used by the robot when replanning actions.

Algorithm 1 Deep Haptic Model Predictive Control

1: **input:** estimation model F, prediction model G, cost function J, time steps between predictions τ_p.
2: $t \leftarrow 0$.
3: **while** force $<$ 10 N and joint limits not reached **do**
4: Observe new sensor measurement $\boldsymbol{x}_t$.
5: **if** $t \bmod \tau_p = 0$ **then**
6: Initialize N action sequences $\{\boldsymbol{a}^{(n)}_{t+1:t+H_p}\}$.
7: Select $\boldsymbol{a}^{*}_{t+1:t+H_p}$ using Equation (2.2).
8: Execute $\boldsymbol{a}^{*}_{t+1:t+\tau_p}$.
9: $t \leftarrow t+1$.

a person's entire arm.

We update the robot's action by selecting the sequence of actions that minimize Equation (2.1). This can be denoted as,

$$\boldsymbol{a}^{*}_{t+1:t+H_p} = \underset{\boldsymbol{a}_{t+1:t+H_p}}{\arg\min}\ J(\boldsymbol{x}_{1:t}, \boldsymbol{a}_{t+1:t+H_p}). \tag{2.2}$$

Algorithm 1 presents our procedure for updating the robot's actions during robot-assisted dressing. At each time step t, we observe sensor measurements x_t. Every τ_p time steps, our controller chooses the actions $\boldsymbol{a}^{*}_{t+1:t+H_p}$ that minimize the cost function, based on a set of N candidate action sequences, $\{\boldsymbol{a}^{(n)}_{t+1:t+H_p}\}$. In this work, we use $H_p = \tau_p = 20$ and we initialize a fixed set of $N = 28$ actions whose velocity lie within a hemisphere facing the +X

global coordinate axis. Computing the cost for each action sequence involves predicting a sequence of future end effector measurements $\hat{\boldsymbol{x}}_{t+1:t+H_p}$ and feeding these measurements into the estimator, F, to estimate the force map at time $t + H_p$. We terminated a trial when the magnitude of forces measured at the robot's end effector exceeded 10 N, or the robot's arm reached its joint limits, which can occur when the arm fully extends to pull a garment onto a participant's shoulder.

This predictive control approach runs in real time on a PR2, using only the robot's on-board CPUs, and both our estimator and predictor can make predictions at $\sim$2 kHz. One limitation is that our model is constrained to relatively short horizon tasks. Notably, our system performed well even with short horizon planning. Computation time limits both the action replanning rate and the prediction horizon, yet our work leaves significant room for future improvements with GPUs, greater parallelization, and off-board computation. Additionally, in this work, we evaluate our model's predictive capabilities, so we restrict our controller from selecting actions that move "backwards". Future implementations could relax this for more freedom while replanning a trajectory.

2.6 Evaluation

We conducted experiments with 10 participants (2 female, 8 male) with approval from the Georgia Institute of Technology Institutional Review Board (IRB), and obtained informed consent from all participants. We recruited able-bodied participants to meet the following inclusion/exclusion criteria: ≥ 18 years of age; have not been diagnosed with ALS or other forms of motor impairments; fluent in written and spoken English. Their ages ranged from 18 to 30 years. A video of our experiments can be found online[1].

We evaluated our model predictive control approach on two robot-assisted dressing scenarios that involve pulling a hospital gown onto a participant's arm: (1) Full arm dressing: the robot must rotate its end effector to navigate around a participant's elbow and pull the

[1]Video: http://healthcare-robotics.com/haptic-mpc

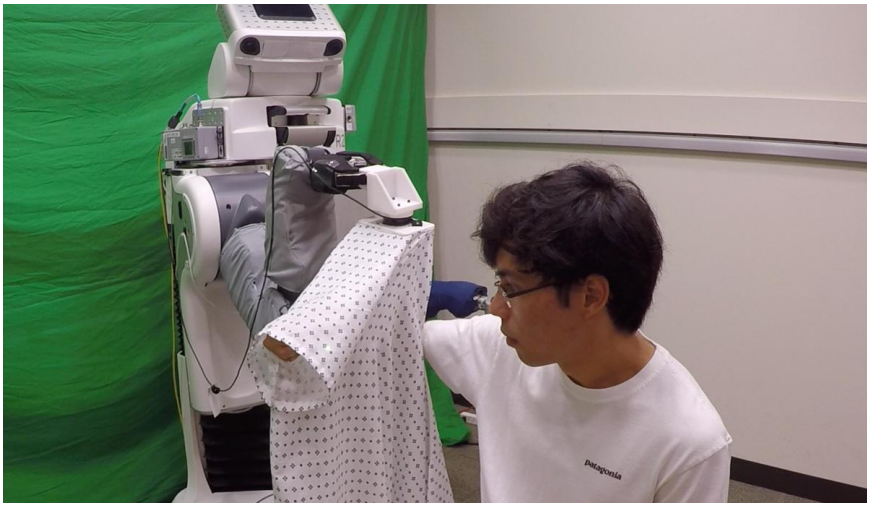


Figure 2.6: The gown getting caught on a participant's fist when the robot uses our MPC approach with a short prediction horizon of 0.01s.

garment onto the person's shoulder, as shown in Figure 2.1. (2) Circumvent a catch: the robot must predict that the garment will soon get caught on a person's fist, as seen in Figure 2.6, and lower its end effector to avoid the catch. The robot performed 24 dressing trials per scenario, for a total of 48 trials per participant. We randomized the dressing scenarios and prediction horizons across all 48 trials. We updated the robot's actions at 5 Hz via Equation (2.2). We selected $w_1 = 0.5, w_2 = 20$, and $w_3 = 0.5$ for our cost function presented in Equation (2.1) as this empirically provided a balance between making task progress and keeping applied forces low.

For each scenario, we tested our method using three different prediction horizons, with 8 trials per horizon: 0.01s, 0.05s, and 0.2s. By testing multiple horizons, we show that a robot can better perform assistive tasks when it can predict the physical implications of its own actions. Note that changing the prediction horizon does not require model retraining since our model is capable of recursively predicting further into the future, which we discussed in Chapter 2.4.

We used a Willow Garage PR2 robot to dress participants. The robot performed actions using the Orocos Kinematics and Dynamics Library[2], which provided joint-level input to

[2]Orocos KDL: http://www.orocos.org/kdl

the PR2's low-level PID controllers. For participant safety, the PR2's arms were compliant and we set low PID gains for all arm joints. We zero out all forces and torques on the ATI force/torque sensor prior to a trial to account for the garment's weight. Additionally, we ran a force threshold monitor that halted all robot movement if forces measured at the robot's end effector exceeded 10 N. All computations to predict force maps for MPC were performed in real time on the robot's on-board CPUs.

Participants sat on a conventional folding chair and we instructed them to hold a specified static posture during each trial, shown in Figure 2.5, and described below:

- Right arm bent 90 degrees at the elbow
- Upper arm and forearm parallel with the ground
- Fingers curled into a fist, knuckles vertically aligned
- Thumb folded inwards over the fingers

We set the initial robot configuration to hold the gown 15 cm in front of the participant's fist with the forearm direction normal to the opening in the gown. All participants started each trial seated comfortably while holding his or her arm in the specified posture. To promote consistency of arm position for appropriately comparing results with different prediction horizons, we used a commercial grade FDA approved laser pointer that pointed at the desired location for the participant's metacarpal-phalangeal joint—the base of the participant's thumb. We placed the laser on an adjustable height table to the left of the participant, facing the robot and orthogonal to the person's forearm, and we aligned the laser according to the participant's height and posture.

We evaluated this work with participants who held a fixed arm pose, yet it may be preferable for a participant to hold their arm in different poses. We note that predicting the future forces applied to a person at varying poses remains an open problem and a limitation of our current work. Allowing the robot to estimate a person's pose prior to dressing, as seen in other works [11, 10], may help alleviate this issue.

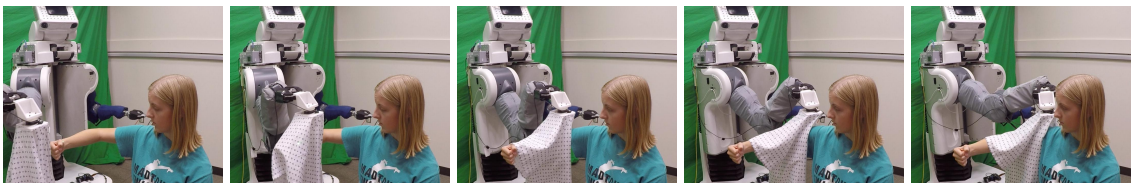


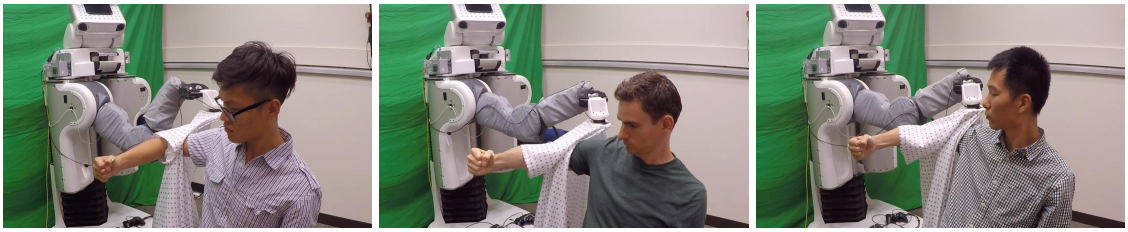
Figure 2.7: (Full Arm Dressing) With our predictive control approach and a horizon of 0.2s, the robot is able to navigate around a person's elbow and pull a gown up to a participant's shoulder.

2.6.1 Full Arm Dressing

For half of the dressing trials, we evaluated the robot's ability to navigate around the elbow and pull the garment entirely up the participant's arm. We were interested in what actions emerged when the controller's primary focus was to minimize the predicted forces applied to a participant's arm.

During a dressing trial, the robot selected actions that minimized the cost from Equation (2.1). Each trial began with the PR2 holding the top of the gown opening 10 cm above the top of a participant's fist. We marked the end of a dressing trial whenever the magnitude of forces measured at the end effector exceeded 10 N, or the robot's arm reached its joint limits. For the full arm dressing trials, we classified a trial as successful if the trial completed without reaching the force threshold and the inner seam on the sleeve, defined by where the sleeve is sewn onto the main body of the gown, had passed the participant's elbow. Figure 2.7 shows a successful sequence of this dressing scenario when the robot used our MPC method with a prediction horizon of 0.2s (20 time steps). Note that once the robot's end effector reaches a person's elbow, the robot can continue to minimize applied forces by performing a yaw rotation to navigate around the elbow and begin moving along the upper arm. This results in the robot pulling the garment entirely up a person's arm.

In Figure 2.8, we display outcomes of dressing trials for the three prediction horizons. For a horizon of 0.01s, the predicted force maps across candidate actions are nearly identical. Because of this, the robot was unable to find an action that significantly lowered applied forces and instead continued to pull the garment into a person's elbow until the



0.01s 0.05s 0.2s

Figure 2.8: (Full Arm Dressing) Example dressing outcomes for each of the three time horizons. The garment gets caught at a person's elbow for a time horizon of 0.01s, but our method successfully rotates the end effector and pulls the gown along a person's upper arm for horizons of 0.05s and 0.2s.

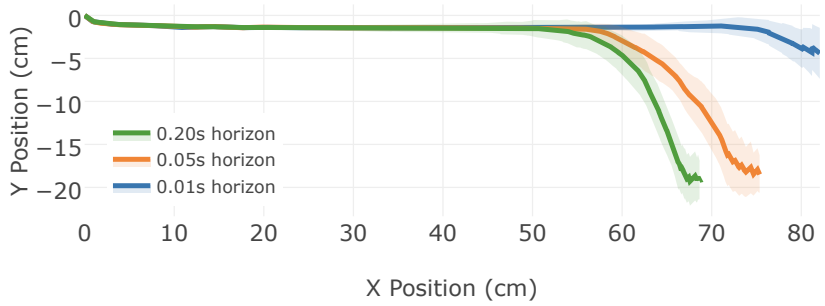


Figure 2.9: Top-down view of the end effector's path for the three prediction horizons during the full arm dressing trials. Paths are averaged across all 10 participants and background shading represents one standard deviation. The robot successfully dressed a person's arm with a 0.05s and 0.2s prediction horizon, yet the robot failed to rotate around the elbow with a 0.01s horizon.

10 N threshold was reached. In contrast, both the 0.05s and 0.2s horizons led to the robot rotating its end effector and pulling the garment up to a participant's shoulder, successfully navigating around the person's elbow. Both Figure 2.7 and the supplementary video show this procedure in detail.

Figure 2.9 shows a top-down view of the end effector path for each prediction horizon, averaged across all 10 participants. A horizon of 0.2s led to the robot rotating and moving along the upper arm sooner than for a horizon of 0.05s, yet both led to actions that fully dressed a person's arm. The task success rates for each prediction horizon can be found in Table 2.1. These success rates are averaged over 80 trials for each scenario. Figure 2.10 displays the magnitude of the force measured at the robot's end effector across trials for all

Table 2.1: Task success averaged over all dressing assistance trials.

	Prediction Horizon		
	0.01s	0.05s	0.2s
Full Arm Dressing	1.25%	97.5%	98.75%
Circumvent a Catch	6.25%	26.25%	97.5%

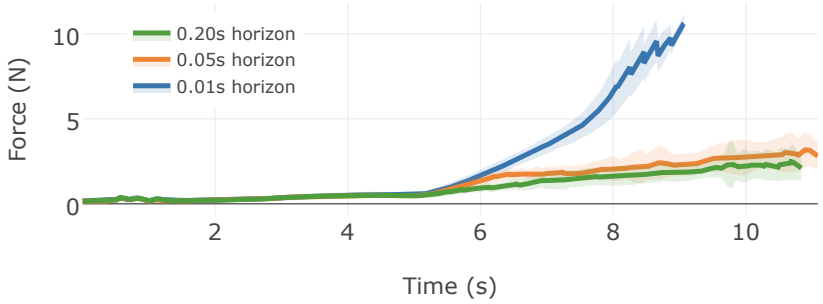


Figure 2.10: Magnitude of force measured by the ATI force/torque sensor at the robot's end effector during the full arm dressing trials. Results are averaged across all 10 participants and background shading represents one standard deviation. The run time for each horizon differs based on when the force threshold or joint limits were reached. With a shorter prediction horizon of 0.01s, the controller applied undesirable high forces at a person's elbow.

10 participants. For a 0.01s horizon, we again notice that the robot continues to apply more force on a person's elbow until it reaches the 10 N threshold.

When contact occurs between the garment and a person's body, our control approach can use haptic and kinematic observations to dress a person by primarily minimizing predicted forces. Yet, a limitation of this purely haptic and kinematic-based approach is that the controller is provided with no information about a person's initial pose. As a result, the robot would be unable to recognize or replan actions if the garment were to entirely miss a person's body. Future work could address this by incorporating other modalities, such as vision-based techniques, to estimate a person's pose before or during dressing [8, 42].

Figure 2.11: (Circumventing a Catch) With a horizon of 0.2s, our approach predicts that the garment will soon get caught on a participant's fist and apply large forces onto the fist. Note how the robot's end effector drops closer to the participant's arm, shown by the red line, to avoid the garment from catching.

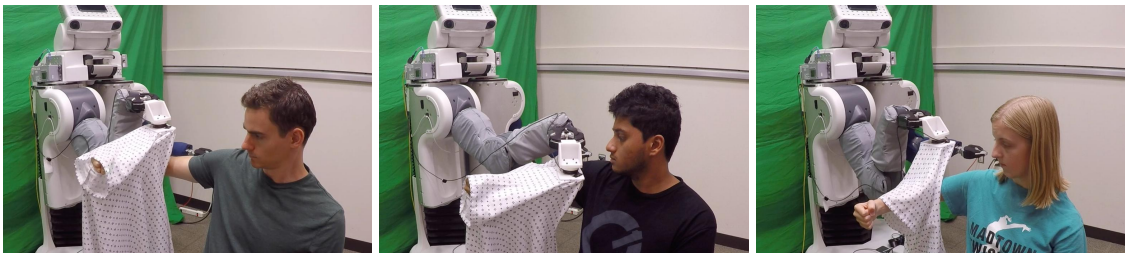
2.6.2 Circumvent a Catch

In this section, we evaluate our model's ability to predict that a garment will get caught and apply large force onto a participant's fist. During these trials, we also evaluated how well our MPC approach selected actions that properly averted the catch in order to reduce predicted forces. We adjusted the starting height of the robot's end effector according to each participant's arm height. Specifically, we aligned the end effector so that the bottom seam of the sleeve would get caught in the middle of a participant's fist when the robot followed a forward linear trajectory. A dressing trial ended whenever the end effector forces exceeded 10 N, or the end effector reached the participant's elbow, along the X-axis. A trial was successful if the end effector reached the elbow along the X-axis without exceeding the force threshold.

Figure 2.11 presents a sequences of images for a successful trial with a 0.2s horizon in which the robot's end effector would drop down closer to a participant's forearm to bypass the catch. Notice that the robot could also choose to lift its end effector to avoid the catch. The robot may not have chosen to lift up over the hand due to the forces that occur when the entire garment drags across a person's fist.

Figure 2.12 shows example outcomes of dressing trials for each of the three prediction horizons. A horizon of 0.01s consistently led to the garment getting caught on a person's fist for 93.75% of the trials, as shown in Table 2.1. A prediction horizon of 0.05s also failed to avoid the catch for most trials.

Finally, Figure 2.13 shows a side view of the end effector path for each prediction



0.01s 0.05s 0.2s

Figure 2.12: (Circumventing a Catch) Example dressing outcomes when our controller attempts to avoid the garment from catching on a participant's fist. The garment often gets caught at a person's fist for horizons of 0.01s and 0.05s, yet our approach successfully navigates around the catch with a 0.2s horizon.

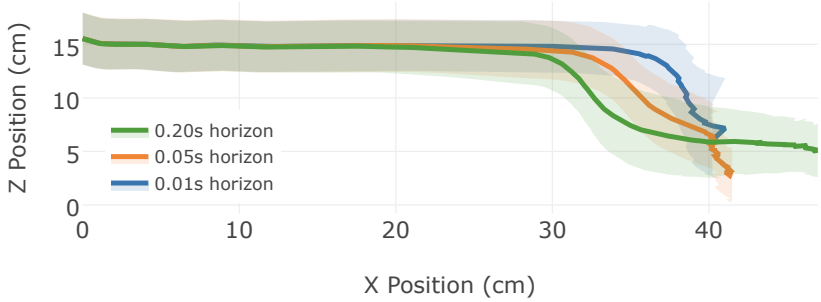


Figure 2.13: Side view of the end effector's path for the three horizons while the robot attempted to circumvent a catch. Paths are averaged across trials from all 10 participants and background shading depicts one standard deviation.

horizon, averaged across all 10 participants. As shown, the horizon length impacts how soon our approach detects the catch and replans. The controller attempted to move the end effector downwards to avert the catch for all three prediction horizons. However, timing is crucial and only the 0.2s horizon allowed our method to detect the catch soon enough to consistently avoid it.

Overall, these results suggest that our approach can enable a robot to predict and react to the forces a garment will exert onto a person during robot-assisted dressing. With a prediction horizon of 0.2s, our model predictive controller is able to fully dress a person's arm in clothing and mitigate the chance of a garment getting caught on a person's body.

2.7 Conclusion

In this work, we presented a learning-based MPC approach that allows a robot to predict the physical implications of its actions and reduce applied force to a person during robot-assisted dressing. We trained a recurrent model on data collected in a self-supervised setting from a physics-based dressing simulation. Unlike prior robot control approaches that use vision-based techniques, our model is able to predict the forces applied to a person's body using only haptic and kinematic measurements from a robot's end effector.

Our model is trained via purely supervised learning, which allows us to define a cost function for MPC post training. This cost function enables a robot to prioritize actions that minimize the predicted force applied to a person's body during physical assistance. Note that this cost function could be changed for different tasks or to allow for personalization, without needing to retrain the model. For a person with a weak or injured wrist, a new function might be defined that primarily focuses on reducing forces applied to the person's hand or wrist. When coupled with state estimation [43, 44], it may be possible to define dynamic cost functions that change depending on the current state of a task.

We evaluated our method with a PR2 that pulled the sleeve of a hospital gown onto the arms of 10 human participants. Our approach enables a robot to predict and react to the forces a garment will exert onto a person during robot-assisted dressing. Our approach also runs in real time on a PR2, using only the robot's on-board CPUs, yet computation time may be a limiting factor for tasks that require faster action replanning rates or longer prediction horizons. From our experiments, we observed emergent behaviors during dressing as we increased the prediction horizon for MPC. With a horizon of 0.2s, our predictive controller was able to fully dress a person's arm in clothing and mitigate the chance of the garment getting caught on the person's body.

CHAPTER 3
TRACKING HUMAN POSE DURING ROBOT-ASSISTED DRESSING USING SINGLE-AXIS CAPACITIVE PROXIMITY SENSING

3.1 Research Summary

Dressing is a fundamental task of everyday living and robots offer an opportunity to assist people with motor impairments. While several robotic systems have explored robot-assisted dressing, few have considered how a robot can manage errors in human pose estimation, or adapt to human motion in real time during dressing assistance. In addition, estimating pose changes due to human motion can be challenging with vision-based techniques since dressing is often intended to visually occlude the body with clothing. We present a method to track a person's pose in real time using capacitive proximity sensing. This sensing approach gives direct estimates of distance with low latency, has a high signal-to-noise ratio, and has low computational requirements. Using our method, a robot can adjust for errors in the estimated pose of a person and physically follow the contours and movements of the person while providing dressing assistance. As part of an evaluation of our method, the robot successfully pulled the sleeve of a hospital gown and a cardigan onto the right arms of 10 human participants, despite arm motions and large errors in the initially estimated pose of the person's arm. We also show that a capacitive sensor is unaffected by visual occlusion of the body and can sense a person's body through cotton clothing.

3.2 Introduction

Robotic assistance provides an opportunity to increase independence and privacy for people with motor impairments, such as some older adults, who need assistance with dressing.

However, providing assistance with dressing remains a challenging task for robots.

Most robot-assisted dressing approaches consider the task of pulling a garment onto a human or mannequin body at a known pose [45, 11, 42], yet few have explored how a robot can correct for errors in the estimated pose of a person, or track human motion during dressing. Estimated pose error can result from either poor initial estimation or movement of the body after the pose was estimated. In addition, dressing often aims to visually occlude a person's body with a garment and the robot's arms can further occlude the body. Because of this, human pose estimation during dressing and tracking pose changes due to human motion is difficult for standard vision-based approaches [9].

In this work, we present a method that uses capacitive sensing to estimate distance to a person's arm in real time, enabling a robot to follow the contours of the person's arm and track the person's motion during robot-assisted dressing. Our capacitive sensing approach can directly estimate the distance to a person's arm using a closed form equation which we derive from the capacitance equation for a standard parallel plate capacitor. Unlike many computer vision approaches for pose estimation, our approach has minimal computational requirements due to the low-dimensional capacitance signal. Furthermore, this approach is rapid, with low latency and high sampling rate, and has a high signal-to-noise ratio. These qualities allow a robot to be highly responsive to both error and human motion during robot assistance when using just a simple PD controller.

We evaluated our method with 10 human participants in a study during which a robot pulled the sleeve of both a hospital gown and a knitted sweater (e.g., a cardigan) onto the participant's arm, as shown in Figure 3.1. We show that a robot can use our capacitive sensing method to adapt to errors in human pose estimation and successfully dress a person's arm despite large error. We also demonstrate how our method allows a robot to track human motion even when a person's arm is already covered by fabric clothing. In addition, capacitive sensing can also be used to detect when contact occurs between a robot and the person receiving assistance, which we discuss in Chapter 3.5.3. From our evaluation, we

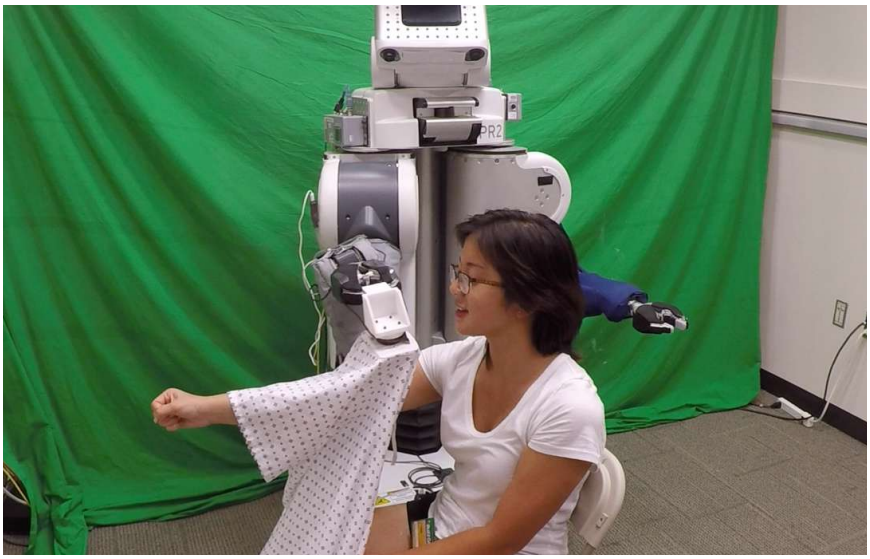


Figure 3.1: A PR2 pulling a hospital gown onto a human participant's arm. The robot uses our capacitive sensing method to adjust for pose estimation errors and human movement during dressing.

found that capacitance measurements had low variability across all participants, which suggests that our capacitive sensing approach may generalize well when providing assistance to a wider population of people.

In this chapter, we make the following contributions:

- We propose and evaluate a capacitive sensing approach for regulating the distance between a robot's end effector and a person's body.
- We show that our capacitive sensing approach can enable a robot to follow the contours of a person's arm even during unscripted arm movement.
- We provide evidence that the capacitance to distance measurements are consistent across people and do not require individual calibration.
- We demonstrate that this approach is able to follow the contours of a person's arm even when the person is already wearing cotton clothing that may cause visual occlusions, as can occur when assisting with an outer layer of clothing.

3.3 Related Work

3.3.1 Robot-Assisted Dressing

Several researchers have considered robot-assisted dressing, yet almost all previous work assumes that a person will hold a stationary pose while being dressed. In addition, few have considered how a robot can adjust for pose estimation errors that occur after dressing begins. We observe this to be an important issue facing prior robot assistance research for other assistive tasks, where human users would often change their pose between when the pose was estimated and the task was completed [46]. One area of work has considered the use of visual information to determine a person's initial pose before dressing. Klee et al. used a vision system to detect a person's pose before a Baxter robot assisted in putting on a hat [8]. Their vision system ensured that a person held the correct stationary pose and was used to halt robot movement if human motion was detected.

Another research area has focused on visual features for tracking the state of a garment during dressing. Koganti et al. used RGB-D and motion capture data to estimate the topological relationship between a garment and a person's body [28]. Tamei et al. used a motion capture system to estimate the topological relationship between a garment and a stationary mannequin [7]. They then had a dual-arm robot learn to pull a T-shirt over the mannequin's head using reinforcement learning. Similarly, Yamazaki et al. used a depth camera to determine a trajectory for a humanoid robot to assist in pulling up a pair of trousers and they used optical flow to estimate the cloth state [10]. However, these works all assumed the person remains stationary while being dressed.

Pignat et al. used a Baxter robot to pull one sleeve of a jacket onto a person's arm [29]. The robot tracked the person's hand movement in real time using an AR tag and the robot began dressing once the person placed his or her hand in the sleeve entry of the jacket. They did not explicitly model nor account for errors or human motion after the robot began dressing a person. Similarly, Chance et al. used a Baxter robot to determine dressing errors

and clothing types when dressing a human participant [47]. They varied the initial pose of the human participants. Their robot followed a predetermined trajectory based on the person's initial pose and they did not consider limb movement during dressing. Erickson et al. used a physics-based simulation to estimate the forces applied to a person's body during robot-assisted dressing using haptic and kinematic measurements at the robot's end effector [30]. Their system accounts for translation and rotation of the human limb before dressing, but does not model human motion during dressing assistance. In previous work, Kapusta et al. observed that some starting heights of linear trajectories along the arm may fail and rapidly cause high forces on the person's arm [48]. In this work, we show that our method enables a robot to avoid some failure cases by adapting in real time to the estimated pose of the arm without applying high forces.

Unlike most past work, which has assumed a person remains stationary when being dressed, our research focuses on sensing and adapting to human motion during dressing assistance. Related to human motion and pose estimation, Gao et al. used a Baxter robot to assist a person in putting on a sleeveless jacket, wherein the person would push their arms through the sleeves [9]. Despite this, the robot followed a fixed trajectory given by the person's initial pose which was estimated using a top-view depth camera. As the researchers noted, *the depth sensor could not be used to determine a person's pose during dressing because the robot's arms would block visual sight of the body*. Similarly, Gao et al. has considered how a robot can adjust for human movement during dressing using force feedback control [11]. They used RGB-D data to estimate a person's pose and compute an initial robot trajectory for dressing a sleeveless jacket. They then proposed a stochastic path optimization approach for personalized robot-assisted dressing that locally adjusts the robot's motion based on force feedback from the robot's end effector. However, force feedback allows a robot to react to human motion only *after* the robot begins applying forces to the person's body. We propose a capacitive sensing approach that enables a robot to directly estimate human motion in real time and track a person's movement before contact

is made or forces are applied.

3.3.2 Capacitive Sensing

Capacitive sensors have been widely used in robotics for contact-based tactile sensing applications. For example, work by Muhammad et al. used capacitive sensing for texture recognition by scanning over a surface and comparing frequency spectrums of sensor measurements [49]. Another area of work has focused on integrating capacitive sensors within artificial skins for obtaining tactile measurements upon contact [50, 51, 52]. These tactile measurements can include information such as applied pressure, force, or strain. Schmitz et al. developed a capacitive skin for contact pressure sensing and they implemented their sensors onto the bodies of three humanoid robots [53]. Phan et al. used a capacitive sensing robotic skin for impact monitoring and force reduction [54]. In addition, Ji et al. designed a capacitive robotic skin for capturing high sensitivity force measurements [55]. In comparison to contact-based sensing, we focus on proximity sensing to measure the distance between a robotic end effector and a person's body.

Capacitive proximity sensors have been broadly explored within human-robot interaction. For example, capacitive proximity systems have been implemented as safety mechanisms that enable a robot to avoid collisions with humans and objects [56, 57, 58]. Kirchner et al. used capacitive sensors mounted on a robotic arm for both proximity sensing and material recognition by also comparing measurements taken at different sensor drive frequencies [59]. Lee et al. designed a capacitive sensor that provides both tactile and proximity sensing for artificial robotic skin [60]. The researchers used a 16×16 grid of capacitive sensors to detect a human hand from 17 cm away. When compared to materials such as metal, plastic, and wood, the researchers found that a human hand resulted in the largest capacitance change as distance between the object and sensor decreased.

In work that closely relates to ours, Navarro et al. proposed the use of capacitive proximity sensing for safer human-robot interaction [61]. The researchers designed a large 3

by 16 grid of capacitive sensors for determining the location and proximity of a human hand with respect to the sensor. Unlike our work, they used a stationary table-mounted sensor with multiple grid cells to sense hand movement, whereas we use a single sensor cell mounted on a robot's end effector to sense and react to human arm motion in real time. Navarro et al. then demonstrated that 2×2 arrays of capacitive sensors mounted on a robot's end effector allowed the robot to follow the curvature of objects, such as aluminum and wooden rods [62]. Unlike our work, the researchers used two parallel capacitive plates, each with four sensors, and assumed a stationary object remained between the two plates. Our work relies on only a single capacitive plate to sense and adjust for human motion during robotic assistance, and we derive an equation for estimating the distance to a person's arm which is analogous to the parallel plate capacitor model.

3.4 Method

3.4.1 Capacitive Sensor Design

To create the sensor's electrode, we covered one side of an 11.5 cm $\times$ 8.5 cm $\times$ 1 mm acrylic sheet in copper foil tape. We connected the electrode to a Bare Conductive Touch Board microcontroller[1], a commercially available board that is designed for filtering readings from a capacitive sensor for touch and proximity-based applications. This board uses a MPR121 capacitive proximity sensor controller that includes high and low frequency noise filtering. We are able to sample capacitance measurements from this sensor at over 200 Hz. We used nonconductive adhesive to mount the electrode to the bottom of a tool held by the PR2's end effector, and we secured the microcontroller to the forearm of the PR2. This tool contains an ATI force/torque sensor and allows the robot to more easily hold a garment. The PR2 is a general-purpose mobile manipulator from Willow Garage with two 7-DoF back-drivable arms and an omni-directional mobile base. Figure 3.2 shows the sensor by itself and mounted on the PR2. In future iterations, these capacitive electrodes could

[1] Bare Conductive: https://www.bareconductive.com/

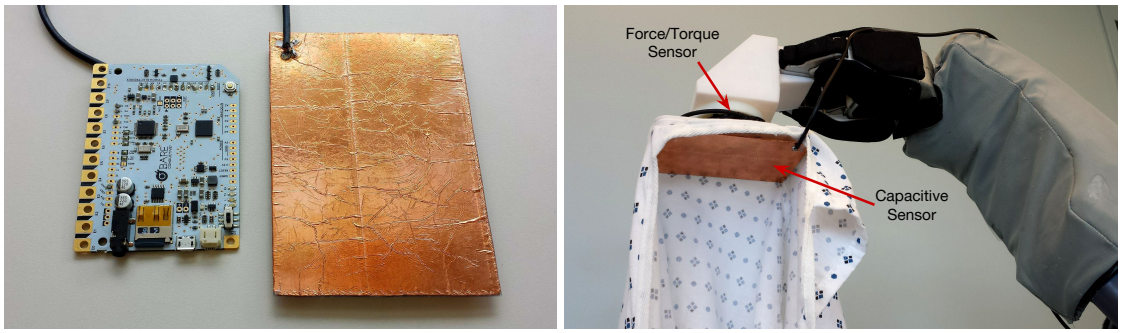


Figure 3.2: Left: Copper plate capacitive sensor and the Bare Conductive microcontroller. Right: The capacitive sensor is mounted to the bottom of a tool that encases an ATI force/torque sensor and holds the hospital gown.

be implemented directly into the front or backside of a robot's fingertip. The lack of sensitivity to cloth between the sensor and a human body, which we explore in Chapter 3.5.3, also presents other possibilities for electrode placement.

3.4.2 Estimating Distance to a Human Arm

Given measurements from the capacitive sensor during dressing, our method estimates a distance between the robot's end effector and a person's arm. To accomplish this, we first collected capacitance measurements when the capacitive sensor and the robot's end effector were at various distances from a person's arm. Specifically, we had the robot position its end effector at six equally spaced locations along the person's arm, from fist to shoulder. At each location, the robot's end effector would start 15 cm above the person's arm and move 1 cm/s downward while we recorded capacitance readings and positions based on the robot's forward kinematics at 100 Hz. Figure 3.3 displays the resulting graph of distance against capacitance measurements. Empirically, we found that our capacitive sensor had a maximum sensing range of approximately 10 cm from a human arm. Capacitance readings taken at a distance further than 10 cm from the arm appeared to be mostly indistinguishable from one another.

We fit a function to this data to estimate the distance between the robot's end effector and a person's arm given a capacitance measurement taken during dressing. The function

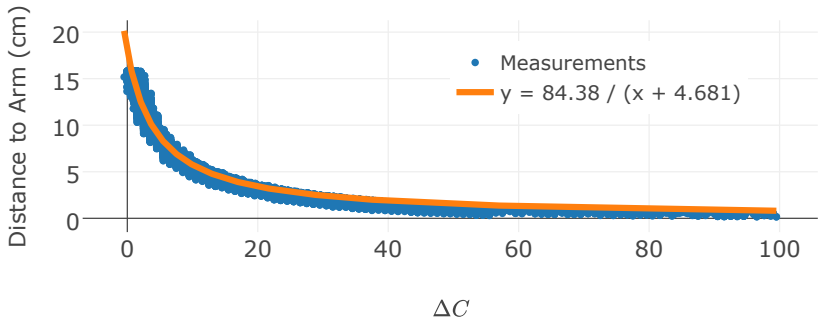


Figure 3.3: Capacitance to distance function fit to sample data along all six human arm locations. The PR2's end effector followed a downward vertical trajectory until contact with six locations evenly spaced along a human arm. The X-axis represents the capacitance change from a baseline defined when no object is underneath the sensor. As discussed in Chapter 3.5.2, we found that this function consistently fit capacitance measurements from all 10 participants in our experimental evaluation.

we fit is inspired by the capacitance equation for a parallel plate capacitor which is defined as,

$$C = \frac{k\epsilon_0 A}{d} \tag{3.1}$$

where C is capacitance, k is the relative permittivity of the dielectric between the plates, $\epsilon_0 = 8.85 \times 10^{-12}$ is the vacuum permittivity, A is the overlapping surface area of the plates, and d is the distance between the plates. We set $\alpha = k\epsilon_0 A$ and reorganize (3.1) as,

$$d = \frac{k\epsilon_0 A}{C} = \frac{\alpha}{C} \tag{3.2}$$

We defined and fit the following function, which is analogous to (3.2) and is defined as,

$$d(\Delta C) = \frac{\alpha}{\Delta C + \beta} \tag{3.3}$$

where ΔC is the change in capacitance from a baseline defined when no object is underneath the sensor. We added a constant β to the denominator to allow for horizontal translation of the curve with respect to ΔC. We fit this function to the data shown in Figure 3.3 using least squares optimization. This resulted in the constants $\alpha = 84.38$ and

$\beta = 4.681$, which achieved a coefficient of determination (R^2) of 0.969. As discussed in Chapter 3.5.2, capacitance versus distance measurements were nearly identical for all 10 human participants, and our evaluation results indicate that this function was accurate for estimating distance to all participants' arms.

3.4.3 Control

We implemented a Cartesian controller on the PR2, which uses the Orocos Kinematics and Dynamics Library[2] to provide joint-level input to the PR2's low-level PID controllers. We ran the Cartesian controller at 10 Hz. For participant safety, the robot's arms were compliant and we used low gains for all arm joints. Additionally, we ran a force threshold monitor that halted all robot movement if forces measured at the robot's end effector exceeded 10 N. We designated the X axis as running along the arm and the Z axis as pointing upward, opposite gravity. At each time step the controller commands the robot's end effector to move 0.5 cm in the X direction, towards the person's shoulder. It additionally commands the robot's end effector to raise or lower in the Z direction with respect to the error in distance from the arm. We can represent our PD controller operating in the Z direction as,

$$u_z(t) = K_p e(t) + K_d \frac{de(t)}{dt} \tag{3.4}$$

where

$$\begin{aligned} e(t) &= d_{desired} - d_{measured} \\ &= d_{desired} - d(\Delta C) \end{aligned} \tag{3.5}$$

K_p and K_d are the proportional and derivative gains, respectively. $e(t)$ is the tracking error: the error between the desired and measured distance from the arm at time t. We manually tuned the controller to be responsive to human motion, selecting $K_p = 0.3$, $K_d = 0.2$, and $d_{desired} = 5$ cm.

[2]KDL: http://www.orocos.org/kdl

3.4.4 Design Limitations

Our approach currently faces a few limitations. To improve sensitivity, our capacitive electrode does not have a ground plate or an active shield, thereby leaving the sensor unshielded from external sources of electromagnetic interference (EMI). Certain clothing types might also impact capacitance measurements, such as those with metal threads or metal decorations. In addition, with only a single large capacitive sensor, we can only estimate the person's arm location along a single direction. Our method has been successful at tracking human motion when error has remained primarily in a single direction, but alterations would be needed to account for pose estimation error and arm movement along multiple directions. For example, sensing 3D human motion, including vertical, lateral, and rotation movements, will likely require three capacitive sensors. Further signal processing, such as low-pass filtering, and learning techniques may also be necessary to resolve interdependencies between multiple nearby sensors.

3.5 Evaluation

We conducted a study with 10 participants with approval from the Georgia Institute of Technology Institutional Review Board (IRB), and obtained informed consent from all participants.

We recruited able-bodied participants to meet the following inclusion/exclusion criteria: ≥ 18 years of age; have not been diagnosed with ALS or other forms of motor impairments; and fluent in written and spoken English. Of the 10 participants, 3 were female and 7 were male. Their ages ranged from 21 to 27 years and their arm lengths ranged from 65 cm to 76 cm.

Every participant started each trial seated comfortably while holding his or her arm in a specified posture, namely: arm straight, pointed forward, parallel to the floor and perpendicular to the front of the robot base; fingers curled into a fist; knuckles vertically aligned.

Figure 3.1 shows this configuration. We recorded the height of a participant's fist and provided this height visually as a reference point before each trial so that the participant could position his or her arm at a consistent height. A supplementary video of our experiments is available online[3].

3.5.1 Error in Human Pose Estimation

As detailed in Chapter 3.3, past work in robot-assisted dressing has relied on visual pose estimation to estimate how a person's arm is oriented prior to pulling on a jacket. However, a person's true arm pose may differ from what was initially estimated, due to visual pose estimation error, modeling error, human movement after the arm pose was estimated, or other factors. We show that our approach using feedback control with capacitive sensing can enable a robot to adapt to even large errors of 15 cm in the estimated pose of a person's arm.

We compared our approach against the robot performing open-loop movements for which the robot did not adjust the vertical height of its end effector using capacitive sensing. When dressing a human participant, the robot's end effector started at a predetermined height above the person's hand and then followed a trajectory approximately parallel to the axis of the person's arm in order to pull on a hospital gown. For each participant, the PR2 performed 10 dressing trials at four predetermined heights: 5 cm, 10 cm, 15 cm, and 20 cm above the person's fist, for a total of 40 dressing trials. We used $d_{desired} = 5$ cm, so the initial pose estimation error for the four heights were 0 cm, 5 cm, 10 cm, and 15 cm respectively. For half of the 40 trials, the PR2 moved the gown in a linear trajectory along the axis of the arm without feedback from the capacitive sensor. For the other half, the robot used capacitive sensing to measure and adjust for the vertical offset between its end effector and the person's arm. When using the capacitive sensor, the robot aimed to keep its end effector 5 cm above the person's arm throughout the entire dressing trial. We alternated

[3]Video: http://healthcare-robotics.com/cap-prox

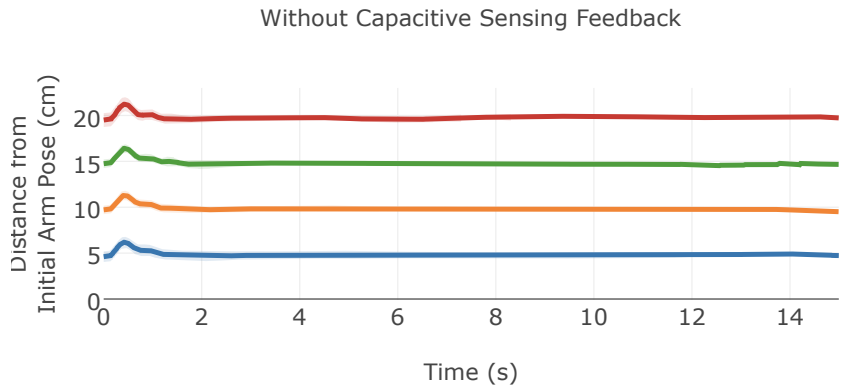


Figure 3.4: Distance of the robot's end effector from the person's arm during dressing, estimated from PR2 forward kinematics and starting height of the person's fist. Colored lines represent trials with a different initial distance between a participant's hand and the robot's end effector before dressing begins. Results are averages from 50 dressing trials: 5 trials at each height for each of the 10 participants. When capacitive sensing feedback is not used, the robot follows a predefined straight path. There is 1.2 cm ($\pm$0.5 cm) of error that occurred at the start of each trial, which may be attributable to the low gains of the PR2 and the outstretched arm at the start of a trial.

methods for the 10 trials at each height.

During each dressing trial, the robot measured the forces and torques at its end effector, the position of its end effector, and capacitance readings. The robot halted all movement if the magnitude of the total force measured by the ATI force/torque sensor exceeded 10 N. We collected all measurements at a frequency of 100 Hz. Figure 3.4 shows the vertical position of the robot's end effector during the dressing trials when capacitive feedback is not used. Note that the reported positions are with respect to the initial height of the robot's end effector, which is offset from the top of a participant's fist, as seen in Figure 3.5. For cases in which the robot's end effector began 5 or 10 cm above a participant's fist, the open-loop trajectory succeeded in pulling the sleeve of the gown up a person's arm. When starting 15 cm or 20 cm above the hand, the open-loop trajectory often led to the arm getting caught in the sleeve, or the arm missing the sleeve entirely.

In comparison, when using capacitive sensing, the end effector was able to adjust for pose estimation errors by moving downward, closer to the participant's arm. Figure 3.6 presents the estimated distance and standard deviation between the end effector and a par-

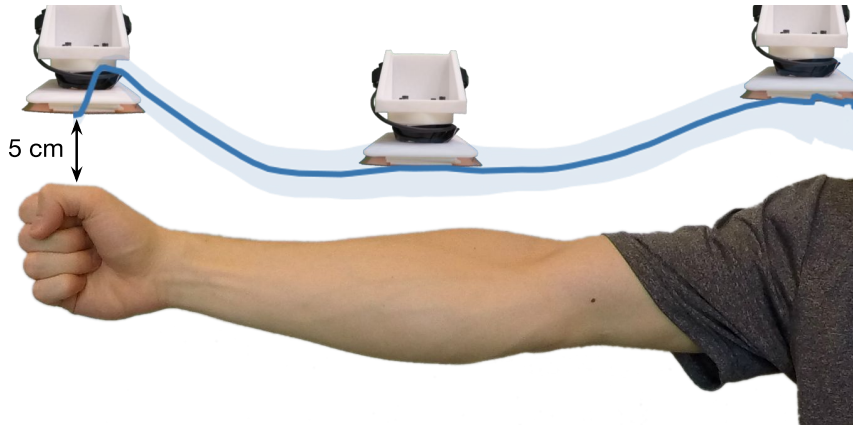


Figure 3.5: With our method, the robot's end effector is able to follow the contours of a person's arm during dressing. We overlay the average movement trajectory taken by the robot when using capacitive sensing feedback to dress all 10 participants. Background shading represents one standard deviation.

ticipant's arm when using capacitive feedback control. These distance estimates are averaged over all 10 participants and were computed from capacitive sensor measurements as shown in (3.3). The figure shows that distance estimates between the robot's end effector and a participant's arm remain between 4 to 6 cm for most of a dressing trial. Furthermore, these capacitive sensing estimates enable the robot to track the contours of a human arm, which is depicted in Figure 3.5. However, estimates are less accurate at distances exceeding the 10 cm maximum sensing range and at the start of the trials. Error at the start of trials may be due to electromagnetic interference (EMI) between the sensor cable and the robot that occurs when the robot's arm is fully extended outwards. Active shielding may help reduce these sources of interference in the future. As shown in Figure 3.4, a bump in the trajectory also occurred at the start of each trial, which may be attributable to the low gains of the PR2 and its outstretched arm at the start of each trial.

3.5.2 Generalization of Capacitance Measurements

We presented an optimized function in Chapter 3.4.2 to estimate the distance between the robot's end effector and a person's arm, given a capacitive sensor measurement. This function was fit to measurements taken from a single person's arm. In this section, we inves-

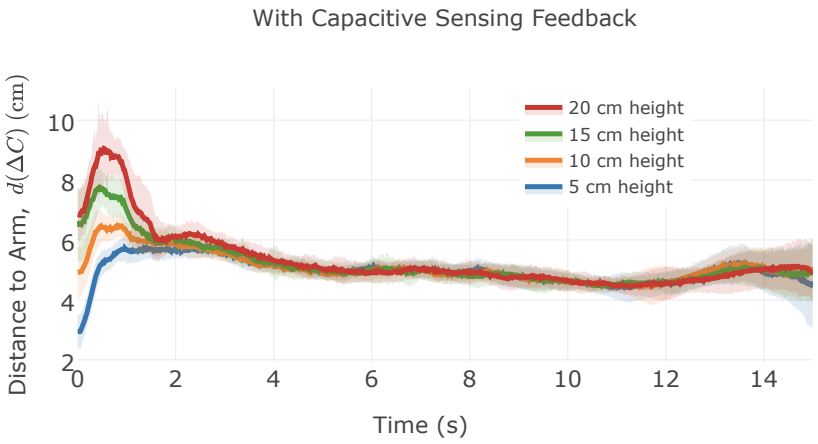


Figure 3.6: Estimated distance measurements between the capacitive sensor and a human participant arm. Distance results are computed as shown in (3.3) and are averaged across 50 dressing trials for each starting height. Shaded regions represent one standard deviation.

tigate the variance in capacitance measurements across different people and we evaluate how well the optimized function generalizes across individuals.

For each participant, we measured capacitance signals when the robot's end effector started 15 cm above the participant's hand and moved 1 cm/s downward until contact and then back upward at 1 cm/s. Figure 3.7 displays these capacitance measurements for all 10 participants. The standard deviation in distance between the robot's end effector and a participant's arm is less than 0.2 cm when averaged across all capacitance measurements. Furthermore, we overlay the optimized function from Figure 3.3 which achieves an R^2 value of 0.964 and is nearly identical to the $R^2 = 0.969$ achieved when the function was fit to measurements from a single person. Overall, we found that capacitance measurements and the optimized function were consistent across participants.

Kirchner et al. and Lee et al. found that capacitance measurements taken over the human body are easily identifiable from many other materials found in human environments [59, 60]. To further validate this, we collected measurements when the capacitive sensor made contact with a fabric gown that was resting on a plastic table. These measurements of both the table and gown are depicted in Figure 3.7 by the green curve. For distances less than 10 cm, these measurements are noticeably different from measurements

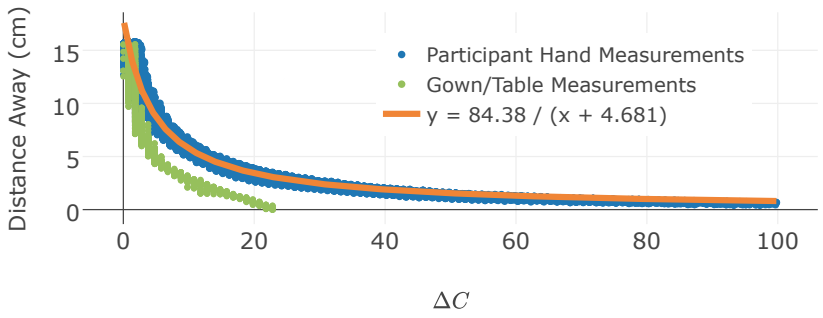


Figure 3.7: Capacitance measurements taken when the PR2's end effector followed a vertical trajectory above each participant's hand. This process is similar to that presented in Figure 3.3, yet is performed with all 10 participants to evaluate variance in capacitance measurements across people. We overlay the function fit to data from Figure 3.3, which achieves an R^2 value of 0.964 on measurements from all 10 participants. The green curve represents measurements captured when the capacitive sensor makes contact with a hospital gown that was resting on a plastic table.

over a human arm, and this aligns with findings by both research groups.

3.5.3 Adjusting to Arm Movement

To illustrate our system's ability to adapt to human motion during robot-assisted dressing, we conducted three demonstrative trials in which each participant moved his or her arm vertically during dressing. We instructed participants to tilt their arms at any time during the dressing process such that their hands remained within 20 cm above or below the initial, horizontal pose. The robot's end effector started 5 cm above the person's hand and moved approximately parallel to the axis of their arm. Figure 3.8 shows a series of images illustrating the movements of one of the participants and the PR2 adjusting to those movements. We observed that by using capacitive feedback control, the robot successfully pulled the gown onto every participant's arm, while tracking the vertical change in the person's arm pose during dressing. Figure 3.9 shows how the vertical position of the robot's end effector changes over time with respect to the estimated height of a participant's arm during a representative dressing trial. For clarity, Figure 3.9 displays only a single dressing trial, yet we note that these results remain consistent across trials from all 10 participants.

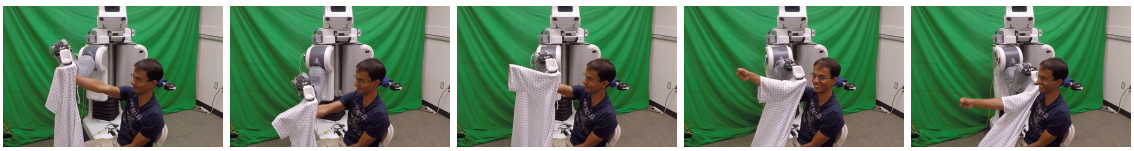


Figure 3.8: The PR2 uses our method to track human movement in real time during dressing assistance with a hospital gown.

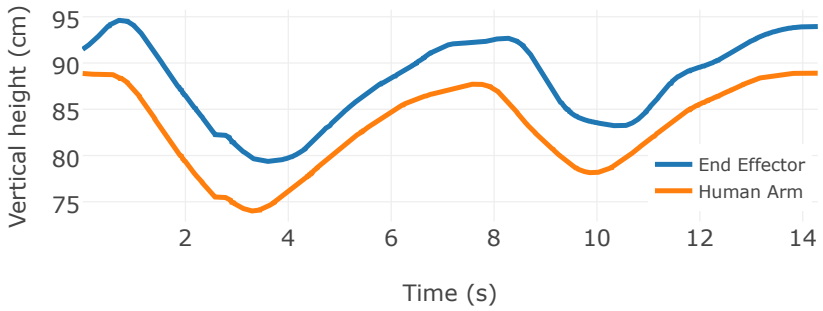


Figure 3.9: Vertical position of the robot's end effector and human arm as the participant moves his/her arm during dressing. This plot shows a representative trial from a randomly selected participant during the gown dressing scenario, as seen in Figure 3.8. The vertical position of the robot's end effector is measured from forward kinematics while the vertical height of a participant's arm is estimated from capacitance measurements.

An additional feature of the capacitive sensor is its ability to detect contact between a person and the robot when $d(\Delta C)$ is arbitrarily close to zero. Given this, we can analyze any contact that occurred with the robot as each participant moved his or her arm during dressing. We detect contact when $d(\Delta C) < 0.5$ cm, which occurs when $\Delta C \geq 165$. During the trials with all 10 participants, the robot made contact with only two participants. For the first participant, contact occurred twice, the first lasting for $\sim$0.1s and the second lasting for $\sim$0.4s. The forces measured at the end effector during these collisions never exceeded 5 N. For the second participant, contact lasted $\sim$0.4s, yet no more than 0.5 N of force was measured. We note that even when collision occurs due to human motion, feedback control with capacitive sensing helps to ensure that contact duration is short and applied forces remain low.

During the next trial, we explored the capacitive sensor's ability to sense a person's body through clothing. For example, this may be applicable when helping dress a jacket onto someone who is already wearing a shirt. During this trial, each participant wore a

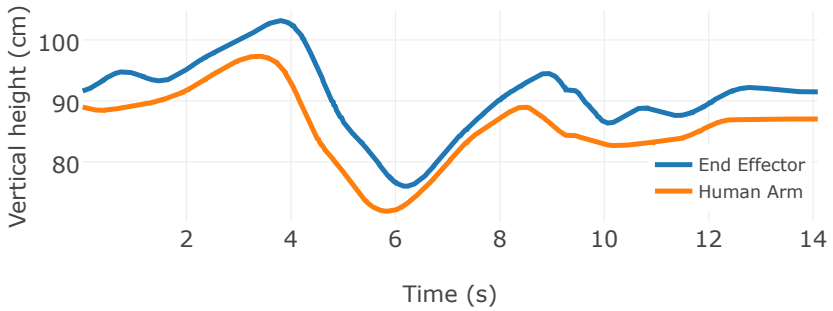


Figure 3.10: Vertical position of the robot's end effector and human arm as the participant moves his/her arm while wearing a long sleeve shirt. This plot shows a representative trial from a single participant during the gown dressing scenario seen in Figure 3.11. We observe that the capacitive sensor is able to track a person's arm movement, regardless of whether the person's arm is covered by clothing.

Figure 3.11: The capacitive sensor tracking a person's arm through clothing. The PR2 is able adapt to a participant's movement even when the participant is already wearing a long sleeve cotton shirt.

long sleeve shirt that covered his or her arm. We again asked participants to vertically tilt their arms as the robot pulled on the sleeve of a hospital gown. Figure 3.10 shows how the vertical position of a participant's arm and the robot's end effector changes while the participant is wearing a long sleeve shirt. In addition, a series of images demonstrating this sequence can be seen in Figure 3.11. Overall, we found that the results closely matched results from when participants' arms were not covered by a long sleeve garment. In both long and short sleeve scenarios, the robot's end effector remained near the target 5 cm range from a participant's arm with an average tracking error of $\sim$1.5 cm despite arm motion. The supplemental video further illustrates how our method can sense and track a person's arm movement, even when the person's arm is occluded by a long sleeve shirt.

Finally, we evaluated how well our approach generalizes to dressing another garment, a long sleeve cardigan. We had the robot pull the long sleeve of the cardigan onto the right arm of all 10 participants, and we encouraged participants to vertically tilt their arms

Figure 3.12: The PR2 pulls a long sleeve cardigan onto a participant's arm while tracking arm movement.

during dressing. The sequence in Figure 3.12 displays the PR2 pulling the cardigan onto a participant's arm while tracking arm movement. Using our method, the PR2 was able to pull the cardigan fully up the arms of all 10 participants. In addition, the PR2's end effector tracked all arm movement, and contact between the end effector and a person's arm occurred only twice. For both occurrences, the contact duration was shorter than 0.4s and the end effector measured less than 2 N of applied force.

3.6 Conclusion

In this work, we demonstrated that a simple capacitive sensor can be used to track human motion and adapt to pose estimation errors during robot-assisted dressing. With a capacitive sensor, our robot estimates the vertical distance between its end effector and a person's arm and we presented a closed form equation for estimating this distance, which enables these estimates to be made in real time with low computational resources. We showed that these capacitive sensors enable a robot to track vertical human motion using only a simple robot controller, and we detailed an approach to extend this to tracking 3D human motion.

A PR2 using our method was able to successfully dress the right arms of 10 human participants with a hospital gown, even when there were 15 cm of initial error in the estimated height of a person's arm. We show that this sensing approach is unaffected by visual occlusion and is able to sense the human body through fabric clothing, such as a long sleeve cotton shirt, which provides evidence that this approach may be useful in other scenarios for which humans and robots physically interact. We further demonstrated that our method allows a PR2 to track human motion and stay within 5 cm of a participant's arm while as-

Figure 3.13: Sleeve differences between the hospital gown and knitted cardigan. The hospital gown has a short and wide sleeve, while the cardigan has a long and narrow sleeve.

sisting with dressing both a hospital gown and a long sleeve cardigan (Figure 3.13). Finally, we found that capacitance measurements remained consistent with low variance across all participants, which provides evidence that this sensing approach may generalize well to a wide population of people.

CHAPTER 4
MULTIDIMENSIONAL CAPACITIVE SENSING FOR ROBOT-ASSISTED DRESSING AND BATHING

4.1 Research Summary

Robotic assistance presents an opportunity to benefit the lives of many people with physical disabilities, yet accurately sensing the human body and tracking human motion remain difficult for robots. We present a multidimensional capacitive sensing technique that estimates the local pose of a human limb in real time. A key benefit of this sensing method is that it can sense the limb through opaque materials, including fabrics and wet cloth. Our method uses a multi-electrode capacitive sensor mounted to a robot's end effector. A neural network model estimates the position of the closest point on a person's limb and the orientation of the limb's central axis relative to the sensor's frame of reference. These pose estimates enable the robot to move its end effector with respect to the limb using feedback control. We demonstrate that a PR2 robot can use this approach with a custom six electrode capacitive sensor to assist with two activities of daily living—dressing and bathing. The robot pulled the sleeve of a hospital gown onto able-bodied participants' right arms, while tracking human motion. When assisting with bathing, the robot moved a soft wet washcloth to follow the contours of able-bodied participants' limbs, cleaning their surfaces. Overall, we found that multidimensional capacitive sensing presents a promising approach for robots to sense and track the human body during assistive tasks that require physical human-robot interaction.

4.2 Introduction

Robots that provide direct physical assistance offer an opportunity to positively impact the lives of many people who require support with everyday tasks. For example, the US Census Bureau has projected that by 2030, over 20% of the US population will be over the age of 65 [63]. Of older adults over 65, $\sim$40% report having some form of disability [64]. People receiving assistance at home from formal caregivers most frequently report receiving assistance with dressing and bathing compared to other activities of daily living (ADLs) and instrumental activities of daily living (IADLs) [64]. Yet, physical human-robot interaction related to these tasks presents several challenges for robots, in part due to difficulties in sensing the human body. Enabling a robot to better estimate human pose and track body motion could be advantageous for many scenarios in which robots and humans physically interact.

In this chapter, we present a technique that uses multidimensional capacitive sensing to estimate the relative pose of a person's limb. Our implementation uses a capacitive sensor mounted on the end effector of a mobile manipulator that consists of 6 conductive electrodes in a 3×2 array. We use a data-driven model that, when given a series of measurements from this capacitive sensor, estimates the relative position of the closest point on the surface of a person's limb, $\boldsymbol{p} = (p_y, p_z)$, as well as the pitch and yaw orientation, $\boldsymbol{\theta} = (\theta_y, \theta_z)$, between a robot's end effector and the central axis of the limb near the point.

When combined with feedback control, we show that this sensing approach enables a robot to track human movement and follow the contours of a person's body while providing assistance with tasks such as dressing and bathing. In addition, our approach provides high sampling rates, with low latency, which allows the robot to estimate a person's local pose at over 100 Hz. Unlike computer vision approaches for estimating a person's pose, our capacitive sensing technique can sense the human body through some materials relevant to assistance, including fabric and wet cloth, which can inhibit a robot's visual sight of

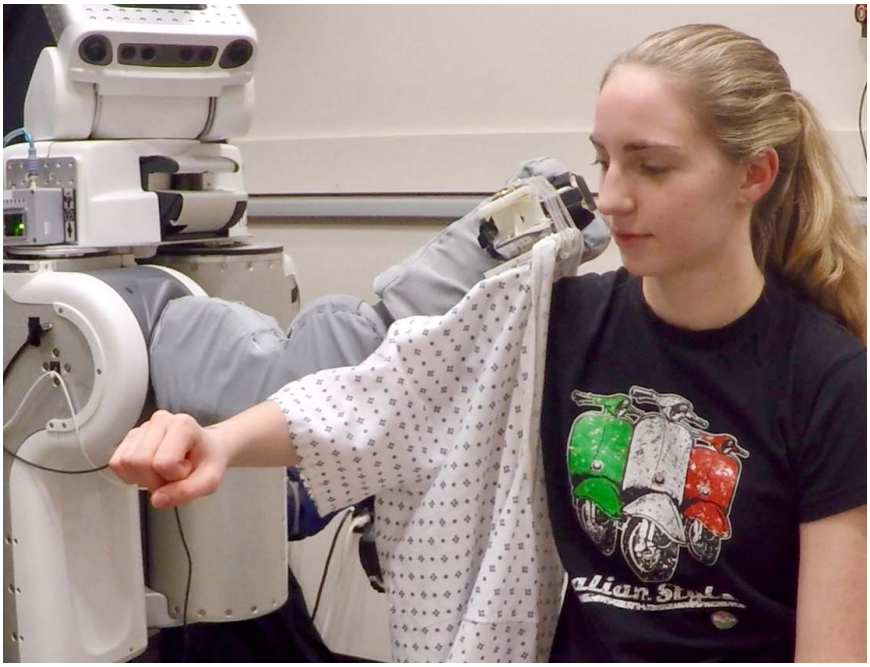


Figure 4.1: A PR2 robot uses multidimensional capacitive sensing to track human motion while pulling a hospital gown onto a participant's arm.

a person during assistive tasks, such as dressing [9]. Furthermore, while current robotic systems have used capacitive sensing at a robot's end effector to follow the 3D contours of static objects [62], no prior works have explored how a robot can use capacitive sensing to track the position and orientation of a dynamic human limb while providing physical assistance.

We demonstrate our approach in a study with able-bodied human participants during which a robot used our capacitive sensing technique to assist in fully dressing the sleeve of a hospital gown onto participants' arms, as depicted in Figure 4.1. We show how our method enables a robot to track human motion, including translation and rotation of a human limb, which allowed our robot to successfully dress participants even in the presence of unscripted human arm motion. Furthermore, we demonstrate how this sensing approach can generalize to assisting with another task—bathing—during which the robot used a wet cloth to clean the outer surfaces of participants' arms and legs, as seen in Figure 4.2.

Through this work, we make the following contributions:

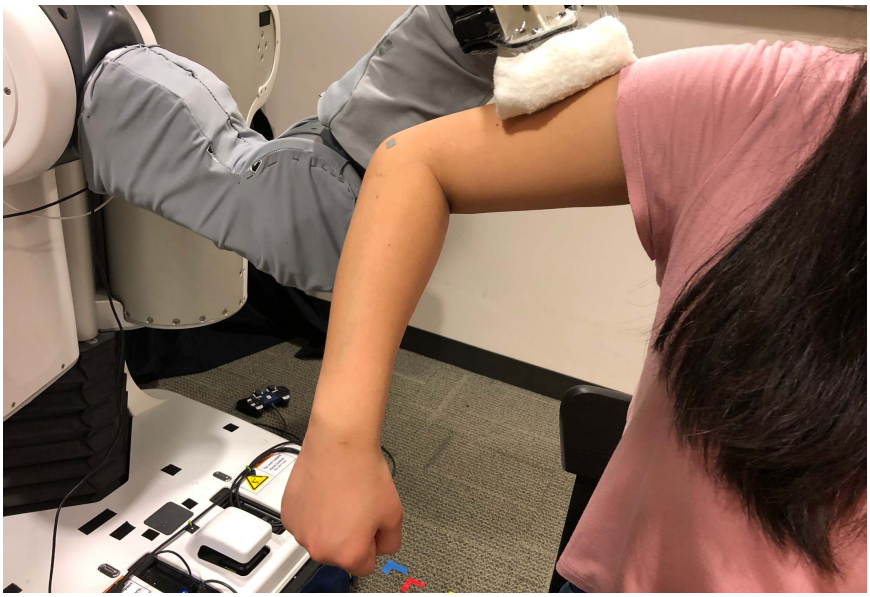


Figure 4.2: A PR2 robot uses our capacitive sensing approach with a wet washcloth to follow and clean a participant's arm.

- We propose and evaluate a capacitive sensor design capable of sensing the local pose of a human limb.
- We present a data-driven model trained on capacitance measurements from a single user and we show that this model can generalize across multiple people to estimate the position of the closest point on a person's limb and the orientation of the limb's central axis.
- When combined with feedback control, we demonstrate how a robot can leverage capacitive sensing to track human motion and assist with two real-world tasks: dressing the sleeve of a hospital gown, and cleaning the arm and leg with a wet washcloth.

4.3 Related Work

4.3.1 Capacitive Sensing

Within robotics, one prominent use of capacitive sensing is for proximity sensing of objects and the human body [59, 65]. Navarro et al. presented a two parallel plate design with 2×2

capacitive sensor arrays that could follow the curvature and orientation of stationary objects that fit in between the two plates [62]. However, this approach does not extend to sensing human limbs larger than the gap between two fixed parallel plate sensors. In comparison, our approach uses a single-sided capacitive sensor array to track dynamic human movement across human limbs of varying shapes and sizes.

Researchers have also looked at capacitive sensing for improving human-robot interaction. Lee et al. designed a 16 $\times$ 16 array of capacitive sensors which could be used as a robotic skin to sense the presence of a human hand from 17 cm away [60]. Hoffmann et al. mounted capacitive sensors to the surface of two industrial robotic arms, which the robots could use to detect and avoid people in the workspace [58]. Additionally, Navarro et al. presented a capacitive sensor design for safer human-robot interaction, consisting of a table-mounted 3 by 16 grid of sensors for estimating the location and proximity of a nearby human hand [61].

In prior work, we introduced a capacitive sensor for sensing one-dimensional vertical distance between a robot's end effector and a human arm, which enabled a robot to track vertical arm motion while dressing the sleeves of both a hospital gown and a long-sleeve sweater [66]. This prior approach used a model inspired by the capacitance equation for a parallel plate capacitor, $d(C) = \frac{\alpha}{C+\beta}$, which estimated the distance between a sensor and a human limb given a single capacitance measurement, C, and constants, (α, β), obtained through a least-squares fit. However, this model lacks a clear generalization to estimating the pose of a person's limb with multiple capacitive sensor inputs, due in part to crosstalk among multiple electrodes in close proximity to one another. Furthermore, no previous work has demonstrated how a robot can use capacitive sensing to track 4-dimensional human limb pose during physical human-robot interaction.

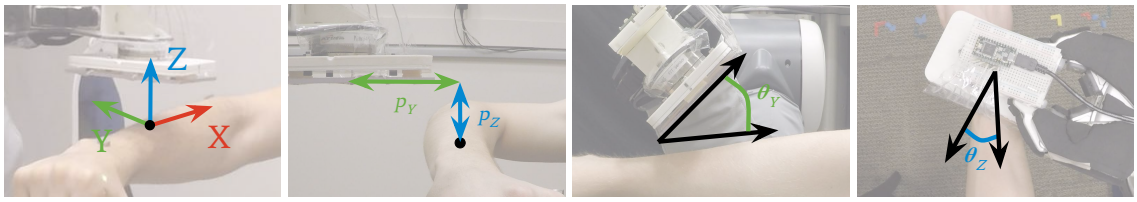
4.3.2 Robot-Assisted Dressing and Bed Bathing

Research on recovering visually occluded human pose in healthcare environments has employed a variety of sensing techniques, including force sensors [11], pressure mats [67], and feature extraction from non-occluded joints in partially occluded images [68]. In comparison to visual feedback approaches, capacitive sensing enables a robot to sense the human body through some opaque materials, such as clothing [66]. When compared to force feedback, capacitive sensing can track human motion *before* the robot applies forces to a person's body.

Many approaches have used vision systems, such as the Microsoft Kinect, to track human pose for robot-assisted dressing [29]. Koganti et al. used the Kinect depth camera and motion capture to estimate the cloth state of a T-shirt as a Baxter robot pulled the T-shirt over a mannequin's head and torso [28]. Chance et al. also used a depth camera with a recurrent neural network to estimate the pose of an occluded human arm as a Baxter assisted in pulling on a jacket [68].

Recent work has also used force sensing to improve robot-assisted dressing performance and reduce the forces applied onto a person's body [69]. Gao et al. proposed a stochastic path optimization approach to adjust a Baxter's motion given force feedback, while dressing a sleeveless jacket [11]. Chance et al. used a 6-axis force/torque sensor to identify dressing errors and clothing types when dressing human participants [47]. In prior work, we presented a deep haptic model predictive control approach for robot-assisted dressing, that enabled a robot to predict the future forces a garment would apply to a person's body given force/torque feedback from a robot's end effector [70].

Relatively few researchers have explored robotic assistance for bathing. Satoh et al. [71] demonstrated how a robotic exoskeleton could help with carrying a person to a bathing location, and Bezarra et al. [72] developed a mechatronic system to control bath water. One common bathing strategy within nursing care are bag baths, which are often performed in bed and do not require large amounts of water or drying [73]. Zlatintsi et al. presented a



(a) (b) (c) (d)

Figure 4.3: (a) The local coordinate frame on the limb at the closest point to the end effector. (b) The lateral and vertical position differences between the end effector and the closest point on the limb. (c) Pitch orientation difference between the end effector and the central axis of the limb. (d) Yaw orientation to the central axis of the limb.

multimodal learning framework for recognizing a person's audio and gesture commands to improve interaction with a robotic bathing system [74]. King et al. presented a robotic bed bathing system that wiped small patches of debris off of a person's static arm and leg [75]. Unlike our approach, the authors used a compliant force-controlled wiping behavior to clean user-selected areas on a person's body and did not estimate human pose or track human motion.

4.4 Sensor, Model, and Training

In this section, we present the multidimensional capacitive sensor design that enables a robot to follow the surface contours of the human body (Figure 4.3) and track human limb movement. In addition, we introduce our data collection process, our model used for estimation, and our robot controller used throughout our experiments with human participants.

4.4.1 Capacitive Sensor Design

Our capacitive sensor design consists of six positively charged capacitive electrodes, which were adhered to an assistive tool held by a robot's end effector. Figure 4.4 displays the capacitive sensor by itself, as well as affixed to the bottom of the tool with a fabric hospital gown or wet washcloth. Each of the sensor electrodes is a 3 cm $\times$ 3 cm square of copper foil mounted in a 3 by 2 grid arrangement. This tool also contains an ATI 6-axis force/torque

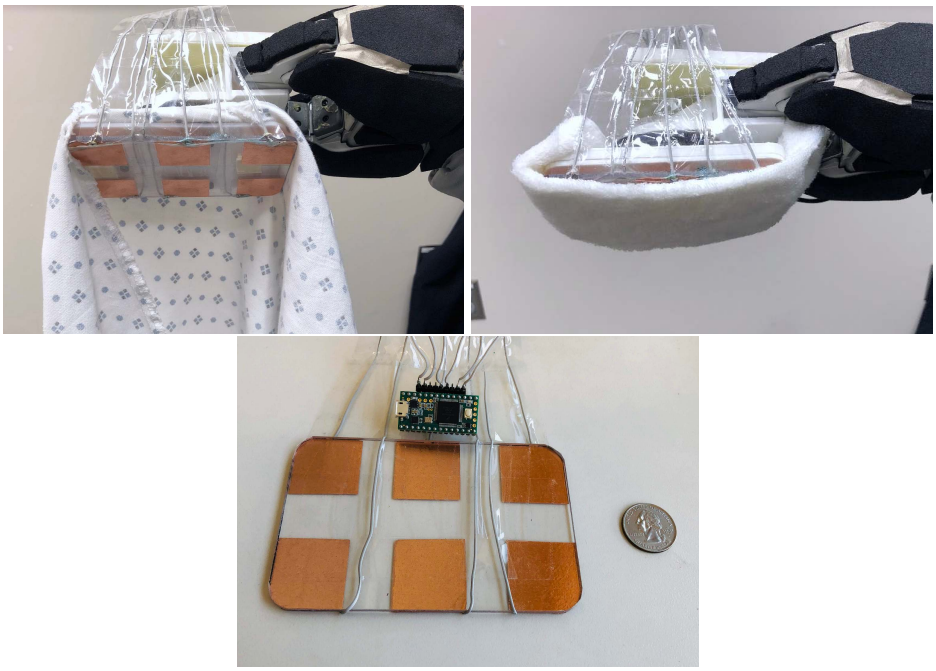


Figure 4.4: Bottom: Six electrode capacitive sensor connected to the Teensy microcontroller board. The capacitive sensor is then mounted onto the bottom of a tool that the PR2 holds. Top left: The tool holding a hospital gown. Top right: The tool with a wet washcloth attached under the sensors.

sensor, which allowed the robot to monitor the force applied to a person's body in the event of direct contact.

We connected the six capacitive electrodes to a Teensy 3.2 microcontroller board, a commercially available board with built-in components for capacitive proximity sensing, including a 1 pF internal reference capacitor. The Teensy is capable of sampling capacitance measurements from all six electrodes at frequencies over 100 Hz.

4.4.2 Estimating Position/Orientation of a Human Limb

Given capacitance measurements from this sensor, we aim to inform a robot about the local pose of a person's limb.

We design a model, $f(\boldsymbol{c}_{t-h+1:t})$, that takes as input a window of prior capacitance measurements, $\boldsymbol{c}_{t-h+1:t}$, from time step $t-h+1$ to the current time t. Our model then outputs, $\boldsymbol{y}_t = (p_{t,y}, p_{t,z}, \theta_{t,y}, \theta_{t,z})$, which includes estimates of the relative position of the closest

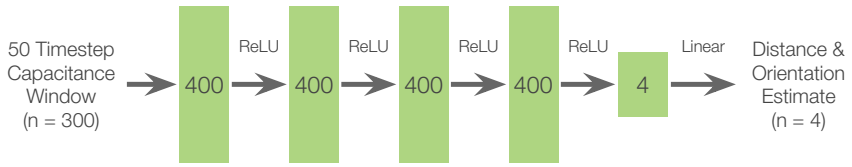


Figure 4.5: Neural network architecture. We apply a ReLU activation after each hidden layer.

point on the surface of a person's limb, $\boldsymbol{p}_t = (p_{t,y}, p_{t,z})$, and orientation of the limb's central axis relative to the sensor's frame of reference, $\boldsymbol{\theta}_t = (\theta_{t,y}, \theta_{t,z})$, as shown in Figure 4.3. We do not estimate the relative position or orientation with respect to the X-axis of a participant's limb, as these two degrees of freedom are difficult to measure for a limb that is approximately cylindrical in shape.

In this work, we use a time window over the last 50 time steps ($h = 50$), or 0.5 seconds given a sampling frequency of 100 Hz. With six capacitance measurements per time step, our input $\boldsymbol{c}_{t-49:t} \in \mathbb{R}^{50\times 6}$ can be vectorized as a 300-dimensional vector, i.e. $f : \mathbb{R}^{300} \rightarrow \mathbb{R}^4$.

To estimate position and orientation, we trained a fully-connected neural network model, with the architecture shown in Figure 4.5. Our model takes as input a 300-dimensional vector of capacitance measurements, $\boldsymbol{c}_{t-49:t}$. Our model, trained in Keras with TensorFlow, consists of four 400 node layers with ReLU activations and a final four node layer which outputs the estimated $\hat{\boldsymbol{y}}_t$. We trained the model for 100 epochs with a batch size of 128. We used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 0.001.

4.4.3 Training a Model

In order to train the network shown in Figure 4.5, we collected time-varying capacitance measurements from a single human participant as a PR2 moves the capacitive sensor around both the person's arm and leg. In Chapter 4.5, we demonstrate how the resulting model can be used by a robot to provide assistance to multiple participants. The PR2 is a general-purpose mobile manipulator, built by Willow Garage, with two 7-DoF back-drivable arms.

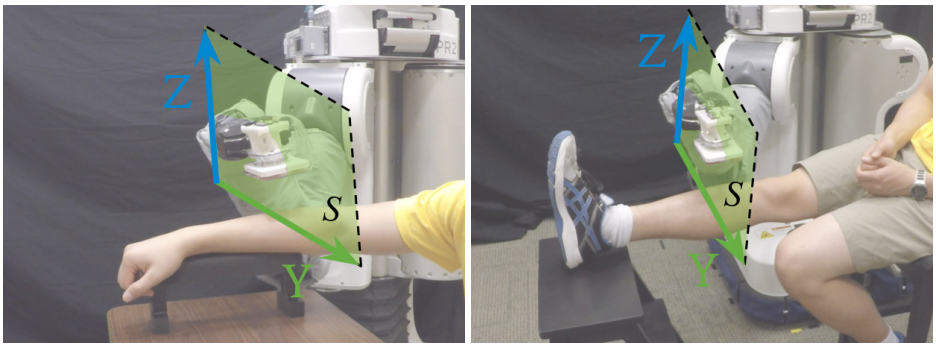


Figure 4.6: The participants elevated their arms and legs parallel to the ground during data collection using an armrest and footstool, respectively. The green highlighted region represents the bounded space of target end effector positions used during data collection.

At each time step t during data collection, we recorded the 50 most recent measurements from the capacitive sensor, $\boldsymbol{c}_{t-49:t}$, as well as the relative position of the closest point on the limb and the orientation of the limb's central axis, $\boldsymbol{y}_t$.

We collected approximately 630,000 data pairs $(\boldsymbol{c}_{t-49:t}, \boldsymbol{y}_t)$ distributed across three locations along the arm (wrist, forearm, upper arm), and three locations along the leg (ankle, shin, knee). We sampled data points at a frequency of 100 Hz as the robot moved around a participant's limb, resulting in $\sim$17.5 minutes of data collection per limb location. As depicted in Figure 4.6, the participant elevated their arm and leg parallel to the ground using an armrest and footstool. Algorithm 2 details our process for collecting training data with the PR2.

We run Algorithm 2 at six positions along the arm and leg. Before data collection begins, we provide the robot with a starting position and orientation above the participant's limb, such that $\boldsymbol{y}_0 = (0, 0, 0, 0)$. We then randomly select a target position $(p_{T,y}, p_{T,z})$ and orientation $(\theta_{T,y}, \theta_{T,z})$ for the end effector from a uniform distribution such that $\boldsymbol{y}_T = (p_{T,y}, p_{T,z}, \theta_{T,y}, \theta_{T,z})$ (line 4), where $p_{T,y} \in [-10\text{cm}, 10\text{cm}]$, $p_{T,z} \in [0\text{cm}, 15\text{cm}]$, $\theta_{T,y} \in [-\frac{\pi}{8}, \frac{\pi}{8}]$, and $\theta_{T,z} \in [-\frac{\pi}{8}, \frac{\pi}{8}]$. The bounded region of possible end effector positions used during data collection is depicted in Figure 4.6. We denote this bounded space of all possible positions and orientations away from the closest point on the limb as S. We then select target velocities (lines 5-6) for the end effector's motion as it moves from $\boldsymbol{y}_0$ to $\boldsymbol{y}_T$,

Algorithm 2 Data Collection

1: **input:** time window h, N iterations.
2: $\mathcal{D} \leftarrow \{\}$
3: **for** $i = 1, \ldots, N$ **do**
4: Select target $\boldsymbol{y}_T = (p_{T,y}, p_{T,z}, \theta_{T,y}, \theta_{T,z}) \in S$.
5: Select velocities $\boldsymbol{v}_p = (v_{p_y}, v_{p_z})$ s.t. $||\boldsymbol{v}_p|| \in [3, 10]$.
6: Select velocities $\boldsymbol{v}_\theta = (v_{\theta_y}, v_{\theta_z})$ s.t. $||\boldsymbol{v}_\theta|| \in [\frac{\pi}{20}, \frac{\pi}{8}]$.
7: $t \leftarrow 0$.
8: **while** $\boldsymbol{y}_t \not\approx \boldsymbol{y}_T$ **do**
9: Observe capacitance measurements $\boldsymbol{c}_t \in \mathbb{R}^6$.
10: Observe $\boldsymbol{y}_t$ using forward kinematics.
11: **if** $t \geq h$ **then**
12: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{c}_{t-h+1:t}, \boldsymbol{y}_t)\}$
13: Take action towards y_T with velocities $\boldsymbol{v}_p, \boldsymbol{v}_\theta$.
14: $t \leftarrow t + 1$.
15: **return** dataset $\mathcal{D}$

including translation velocity, $\boldsymbol{v}_p$ in cm/s, and rotation velocity, $\boldsymbol{v}_\theta$ in radians/s.

At each time step, we record the most recent measurement from the capacitive sensors, $\boldsymbol{c}_t$, along with the position of the end effector with respect to the closest point on the limb and orientation of the end effector with respect to the limb's central axis, $\boldsymbol{y}_t$ (lines 9-12). The robot then takes an action along a linear trajectory towards the target end effector position and orientation, with velocities $\boldsymbol{v}_p$ and $\boldsymbol{v}_\theta$, respectively. Finally, we repeat this entire process for a total of $N = 500$ iterations (line 3), resulting in a dataset $\mathcal{D}$ of pairings $(\boldsymbol{c}_{t-h+1:t}, \boldsymbol{y}_t)$ for training our model.

4.4.4 Control

Throughout our experiments in Chapter 4.5, we control the PR2 to follow the path and contours of a person's limb using a high level Cartesian controller. Our controller uses the TRAC-IK[1] inverse kinematics library to provide joint-level input to the low-level PID controllers at each actuator in the PR2's arm. We used low PID gains at each joint so that the robot's arms were compliant for participant safety. In addition, we implemented a force

[1]TRAC-IK: https://traclabs.com/projects/trac-ik/

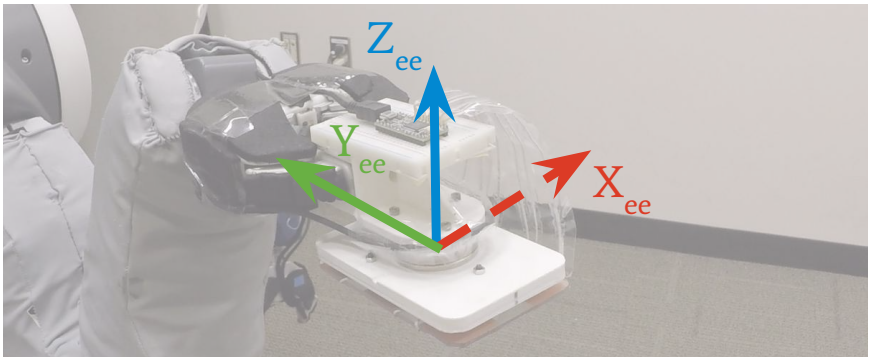


Figure 4.7: The local coordinate frame of the end effector. During our evaluations with human participants, the end effector traverses along the X_{ee} axis at a rate of 2 cm/s.

threshold monitor, which would halt robot movement if the end effector made contact with a participant during an experiment and forces measured at the force/torque sensor exceeded 10 N.

During execution, the controller commanded the PR2's end effector to move 2 cm/s along the X-axis of the end effector, X_{ee}, (see Figure 4.7), approximately along the central axis of the limb. At a time step t, the robot used our trained model to estimate the position of the closest point on a person's limb and the orientation of the limb's central axis, $\hat{\boldsymbol{y}}_t = f(\boldsymbol{c}_{t-h+1:t})$. The controller then computed actions for the robot, $u(t) = (u_y(t), u_z(t), u_{\theta_y}(t), u_{\theta_z}(t))$, representing discrete translational movements $(u_y(t), u_z(t))$ and rotational motions $(u_{\theta_y}(t), u_{\theta_z}(t))$ towards an offset from the closest point on the human limb, $\boldsymbol{y}_{desired} = (p_y, p_z, \theta_y, \theta_z)$. Since our model estimates are with respect to a coordinate frame at the closest point on the person's limb, the robot transforms these actions into the end effector's frame of reference prior to executing the actions. We can also change $\boldsymbol{y}_{desired}$ to adapt to the robot's task at hand. For example, when our robot is providing dressing assistance (Chapter 4.5.2), we define $\boldsymbol{y}_{desired} = (0, 5, 0, 0)$, such that the end effector stays 5 cm above a participant's limb, with the same orientation as the limb. However, when the robot is providing bathing assistance (Chapter 4.5.3), we set $\boldsymbol{y}_{desired} = (0, 1, 0, 0)$ so that the wet washcloth held by the robot makes direct contact with a person's skin. We define our PD controller as,

Algorithm 3 Human Limb Contour Following

1: **input:** estimation model f, control policy u, desired position and orientation for end effector $\boldsymbol{y}_{desired}$, time steps between action updates τ.
2: $t \leftarrow 0$.
3: **while** force < 10 N and end of limb not reached **do**
4: Observe capacitance measurements $\boldsymbol{c}_t \in \mathbb{R}^6$.
5: **if** $t \bmod \tau = 0$ **then**
6: Compute error $e(t)$ using model $f(\boldsymbol{c}_{t-h+1:t})$.
7: Compute action $u(t)$ given control policy in (4.1).
8: Execute $u(t)$.
9: $t \leftarrow t + 1$.

$$u(t) = \boldsymbol{K}_p e(t) + \boldsymbol{K}_d \dot{e}(t) \tag{4.1}$$

where

$$\begin{aligned} e(t) &= \boldsymbol{y}_{desired} - \hat{\boldsymbol{y}}_t \\ &= \boldsymbol{y}_{desired} - f(\boldsymbol{c}_{t-h+1:t}). \end{aligned} \tag{4.2}$$

$e(t)$ is the tracking error, whereas $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are diagonal matrices for the proportional and derivative gains, respectively. We tuned the controller to produce smooth motion while following the contours of a person's limb, resulting in $\boldsymbol{K}_p = \text{diag}(0.025, 0.025, 0.1, 0.1)$ and $\boldsymbol{K}_d = \text{diag}(0.0125, 0.0125, 0.025, 0.025)$. When evaluating the robot's responsiveness to human motion in Chapter 4.5.2, we changed the proportional gains of our controller to $\boldsymbol{K}_p = \text{diag}(0.2, 0.2, 0.1, 0.1)$.

Algorithm 3 details the control process we used throughout our evaluations in Chapter 4.5. During an evaluation trial, the robot continually samples capacitance measurements and updates its actions according to the control policy, u, until the trial completes. The robot observes measurements from the capacitive sensor at 100 Hz (line 4 of Algorithm 3) and runs the high-level Cartesian controller at a rate of 10 Hz ($\tau = 10$). The high-level Cartesian controller uses an updated action to produce new target actuator angles for the low-level PID controllers (line 8). We marked the end of an evaluation trial (line 3) whenever the applied forces measured at the end effector exceeded 10 N, or when the robot's

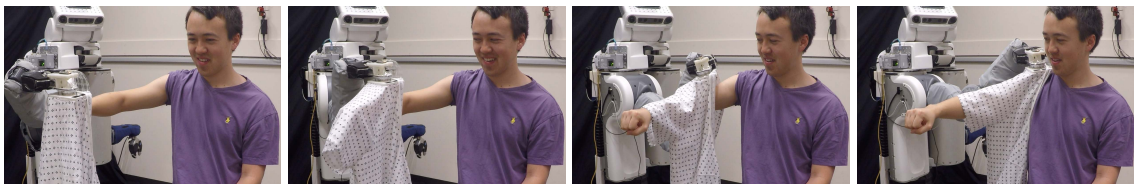


Figure 4.8: The PR2 using capacitive sensing to follow the path of a participant's arm and pull on the sleeve of a hospital gown. Participant's held their arm parallel to the ground plane with a 30 degree elbow bend.

end effector had traversed a participant's entire limb, such as starting above a participant's hand and navigating up to the shoulder.

4.5 Evaluation

In the following sections, we describe how our capacitive sensing approach can enable a robot to assist people with two everyday tasks: dressing, during which a robot pulls the sleeve of a hospital gown onto a participant's arm, and bathing, during which the robot uses a wet cloth to wipe down the surface of a person's body.

To evaluate our capacitive sensor and trained model, we conducted a study with four able-bodied human participants (two females and two males) with informed consent from all participants and approval from the Georgia Institute of Technology Institutional Review Board (IRB). We recruited participants to meet the following inclusion/exclusion criteria: $\geq$ 18 years of age; fluent in written and spoken English; and have not been diagnosed with ALS or other forms of motor impairments. Participant ages ranged from 18 to 27 years with arm lengths between 57-59 cm and leg lengths between 83-90 cm. For each of the assistive dressing and bathing scenarios detailed below, we conducted two trials between the robot and each participant. Our study and experiments can be found in the supplementary video.

4.5.1 Traversing the Human Arm

We began by evaluating the robot's ability to use capacitive sensing to follow the contours of a participant's arm and dress the sleeve of a hospital gown (Figure 4.8).

During a dressing trial, participants held their arm up parallel to the ground with their elbow bent at a 30 degree rotation. The PR2 held a hospital gown in its right end effector and began with the capacitive sensor starting $\sim$5 cm above the center of a participant's hand. The robot then used capacitance measurements to follow the contours of a participant's arm, according to Algorithm 3. We conducted two dressing trials per participant, totalling eight trials across all participants. Figure 4.8 shows the PR2 using the capacitive sensor to follow the path of a participant's arm and pull the sleeve of a hospital gown fully up to the shoulder. Notice that once the robot's end effector reached a participant's elbow, the capacitive sensor sensed a change in orientation between the forearm and upper arm. This resulted in the end effector rotating 30 degrees around the bent elbow to match the orientation of the upper arm. For dressing, we classified a trial as successful if the robot's end effector reached the top of a participant's shoulder with the sleeve around the upper arm. Overall, the robot succeeded in pulling the gown up to the top of a participant's shoulder in all 8 trials.

In order to obtain the ground truth position and orientation of each participant's arm for post-hoc comparison, we placed small infrared reflective markers on a participant's wrist, elbow, and shoulder. We used the PR2's head-mounted Microsoft Kinect depth sensor to record these 3D joint positions, yet these positions were not provided to the robot during a trial. After each trial was completed, we computed the distance between the robot's end effector and the top of a participant's arm at each time step using 3D line segments between these joint positions. Figure 4.9 depicts the Euclidean distance and standard deviation between the robot's end effector and a participant's arm, averaged across all participants. We observed that, on average, the robot's end effector remained within 4 to 6 cm from the surfaces of the participants' arms during the dressing trials.

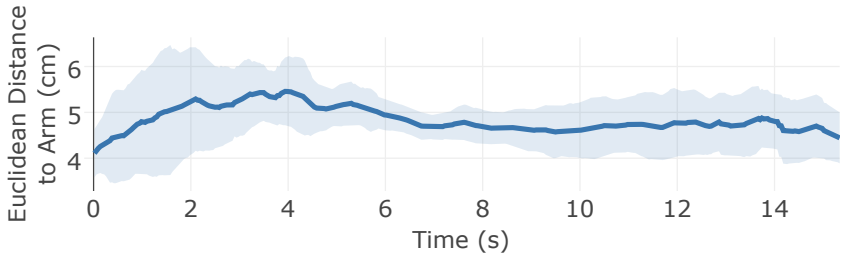


Figure 4.9: Euclidean distance between the robot's end effector and a participant's arm. Given the reflective markers on a participant's joints, we define 3D line segments representing the forearm and upper arm, and we compute distance as the shortest path between the end effector and these line segments. Distance results are averaged across trials from all participants. Shaded regions represent one standard deviation.

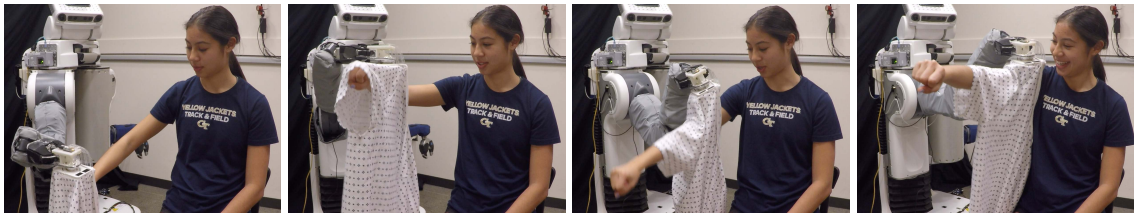


Figure 4.10: The PR2 using capacitive sensing to track vertical arm motion. The PR2's end effector raised and lowered in response to the participant's movement, while also performing pitch rotations to match the orientation of the arm (see third figure in sequence).

4.5.2 Adapting to Arm Motion

We also investigated the robot's ability to use multidimensional capacitive sensing to adapt to human motion during robot-assisted dressing. We first instructed participants to perform vertical arm movements by tilting their arm during dressing, such that their hand remained within 20 cm above or below the starting pose. Participants again started with their arm horizontal to the ground plane, with a 30 degree bend in their elbow. Figure 4.10 shows a sequence of images depicting how the robot adjusted to the vertical motions of a participant. In addition, Figure 4.11 displays how the vertical position of the end effector rose and lowered to match the estimated height of a participant's arm during a single representative dressing trial. Despite arm motions, the robot used capacitive sensing with feedback control to remain on average 4.9 cm above the participant's arm throughout the entire dressing process. We observed that these results were consistent for all participants as the robot

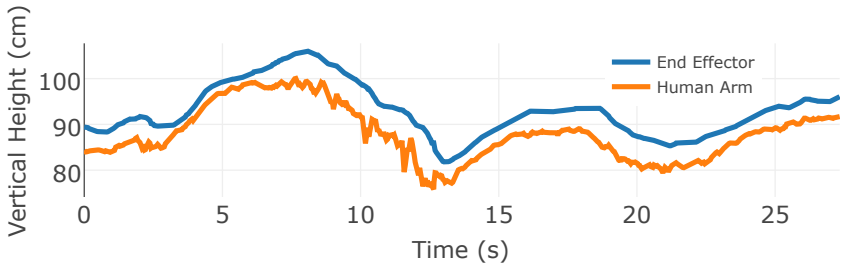


Figure 4.11: Vertical position of the robot's end effector and human arm as a participant moves his/her arm vertically during dressing. For clarity, this plot shows a representative trial from a single participant performing vertical arm movements, as seen in Figure 4.10. The position of the end effector is measured from forward kinematics while the vertical position of a participant's arm is estimated using the capacitive sensor and our neural network model.

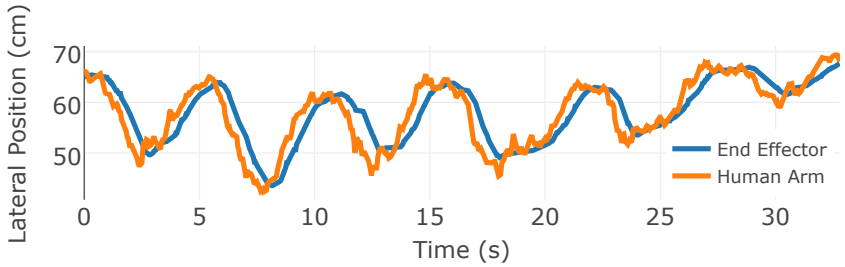


Figure 4.12: Lateral position (along the Y-axis) of the end effector and human arm as a participant shifts his/her arm left to right during dressing. This plot displays a representative trial from a single participant. The lateral position of the end effector is measured from forward kinematics whereas the position of a participant's arm is estimated using the capacitive sensor and our neural network model.

successfully pulled the gown up to each participant's shoulder across all 8 trials.

We then instructed participants to perform lateral arm motions by shifting their arm left to right and bending their elbow, such that their hand remained within 20 cm to the left or right of their starting arm pose. Examples of this vertical arm motion can be found in the supplementary video. Figure 4.12 depicts how the end effector tracked the lateral arm movement for a single participant. On average, the estimated relative lateral position of the end effector, $\hat{p}_y$, remained 2.6 cm from the closest point on the arm throughout the entire dressing trial. These results remained consistent across all participants, as the robot successfully pulled the sleeve up to each participant's shoulder in all 8 trials.

4.5.3 Maintaining Continuous Contact

In this section, we describe how the robot can use this same capacitive sensing approach to assist with tasks that require continuous contact with a person's body, such as bathing. Bathing, and the process of cleaning the body with a wet cloth, is fundamental to bed baths, an assistive task common among people who have difficulty leaving bed to shower or bathe. To assist with bathing, we attached a soft wet washcloth to the bottom of the capacitive sensor, which the robot uses to wipe the surface of a person's body.

Due to the electrical properties of wet cloth, measurements from the capacitive sensor are substantially different from those observed during dressing. As a result, we repeated the data collection and model training procedures, described in Chapter 4.4.3, with a wet washcloth attached to the capacitive sensors. In doing so, we show that our process for training a model on capacitance measurements is both repeatable and can extend to other assistive tasks that require sensing the human body.

We evaluated our capacitive sensing approach with two distinct bathing tasks: cleaning both the arm and the leg of a participant. To better visualize the robot cleaning a limb, we placed a small amount of vibrant blue powder across the limb, which the robot wiped off during a trial. Since the robot makes contact with a participant and the force it applies is distributed across a large soft washcloth, we increased the force threshold in Algorithm 3 to 20 N.

When assisting with wiping a participant's leg, the robot's end effector began above a participant's thigh, and moved downwards towards the ankle. We asked participants to place their leg on a footstool and to bend their knee at a 30 degree angle, which forced the end effector to rotate around the knee and match the orientation of the lower leg to succeed. Figure 4.13 portrays a sequence of images from a successful leg wiping trial. We note that sensing the human body through a wet washcloth poses a challenge for capacitive sensing, yet despite this difficulty, our capacitive sensing approach successfully cleaned all visible blue powder off a participant's leg for 6 of the 8 trials across all 4 participants.

Figure 4.13: The PR2 uses capacitive sensing to follow the path of a participant's leg and navigate around a 30 degree knee bend while wiping blue powder off of a participant's lower leg with a wet washcloth.

Figure 4.14: The PR2 performs a bathing task by using a wet washcloth with capacitive sensing to wipe blue powder off of a participant's arm. The participant's elbow was bent at $\sim$90 degrees and the participant's forearm was tilted 60 degrees downwards towards the ground (see Figure 4.2).

Finally, when the robot assisted with wiping a participant's arm, the robot's end effector began above a participant's hand and moved inwards towards the shoulder. During this scenario, participants held their upper arm parallel to the ground, with a 90 degree elbow bend, and with their forearm and hand tilted 60 degrees downwards towards the ground, as demonstrated in Figure 4.2. An image sequence of the robot wiping a participant's forearm and upper arm can be seen in Figure 4.14. Overall the robot succeeded to clean all visible powder off a participant's arm in 7 of the 8 total trials.

From these image sequences, we can observe that the robot recognized an orientation change at the elbow and rotated to match the estimated orientation of the upper arm. Figure 4.15 and 4.16 present the average force the end effector applied to participants' legs and arms, respectively. We can observe that the washcloth is continually in contact with a participant's limb, while applying less than 6 N of force on average.

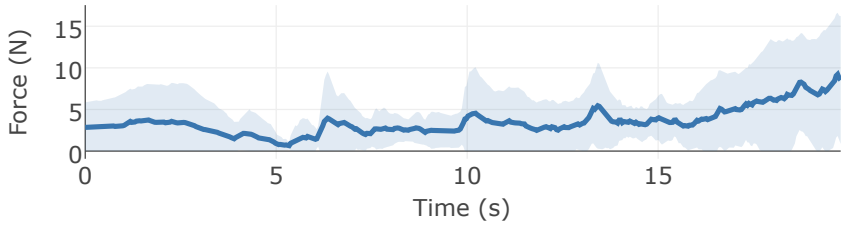


Figure 4.15: Average force applied by the PR2's end effector as the robot cleaned participants' legs. Results are averages across all participants with shaded regions representing one standard deviation.

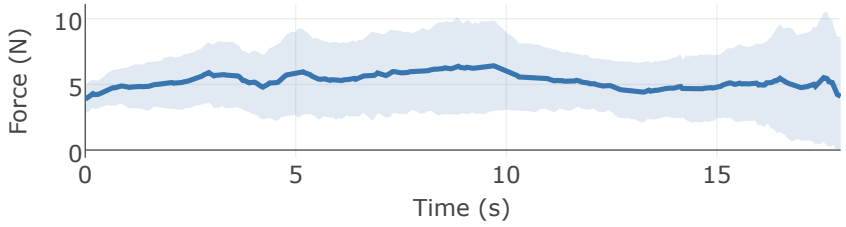


Figure 4.16: Average force applied by the PR2's end effector as the robot cleaned participants' forearms and upper arms. Results are averages across all participants with shaded regions representing one standard deviation.

4.6 Discussion and Conclusion

We presented a multidimensional capacitive sensor capable of sensing the relative pose of a person's limb. Unlike prior control approaches that rely on vision or force feedback, capacitive sensing enables a robot to sense the human body through some opaque materials and track human motion before the robot applies forces to a person's body.

We detailed an approach to train a data-driven model on capacitance measurements from a single participant, which estimates the relative position of the closest point on the surface of a person's limb, as well as pitch and yaw orientation between a robot's end effector and the central axis of the limb near the point.

While our model generalized well to assisting multiple people, there are a few limitations to this approach. First, a model trained on a single person may not always generalize to all people, such as those with very small or large limbs. In these cases, it would be beneficial to train on capacitance measurements collected over multiple people with a wide

variety of limb sizes and shapes. Further, our capacitive sensors do not have active shielding and thus may be susceptible to electromagnetic interference (EMI). Some conductive objects or clothing may also inhibit capacitance measurements, such as jewelry or metal decorations.

We demonstrated how this capacitive sensing approach enables a PR2 to assist with two activities of daily living that many older adults with physical disabilities require assistance with—dressing and bathing. Our approach runs in real time, using only the PR2's on-board CPUs, which enabled the robot to assist with dressing by using capacitive sensing to track human limb movement and follow the contours of participants' arms to pull on the sleeve of a hospital gown. In addition, we demonstrated that our data collection and training process is repeatable and can be used to build customized models for new assistive tasks that require sensing the human body, such as bathing.

Overall, multidimensional capacitive sensing offers a promising approach for robots to sense the human body and track human motion when providing assistance that requires physical human-robot interaction.

CHAPTER 5
CHARACTERIZING MULTIDIMENSIONAL CAPACITIVE SERVOING FOR PHYSICAL HUMAN-ROBOT INTERACTION

5.1 Research Summary

Towards the goal of robots performing robust and intelligent physical interactions with people, it is crucial that robots are able to accurately sense the human body, follow trajectories around the body, and track human motion. This study introduces a capacitive servoing control scheme that allows a robot to sense and navigate around human limbs during close physical interactions. Capacitive servoing leverages temporal measurements from a multi-electrode capacitive sensor array mounted on a robot's end effector to estimate the relative position and orientation (pose) of a nearby human limb. Capacitive servoing then uses these human pose estimates from a data-driven pose estimator within a feedback control loop in order to maneuver the robot's end effector around the surface of a human limb. We provide a design overview of capacitive sensors for human-robot interaction and then investigate the performance and generalization of capacitive servoing through an experiment with 12 human participants. The results indicate that multidimensional capacitive servoing enables a robot's end effector to move proximally or distally along human limbs while adapting to human pose. Using a cross-validation experiment, results further show that capacitive servoing generalizes well across people with different body size.

5.2 Introduction

Physical human-robot interaction and robotic assistance presents an opportunity to benefit the lives of many people, including the millions of older adults and people with physical disabilities, who have difficulty performing activities of daily living (ADLs) on their

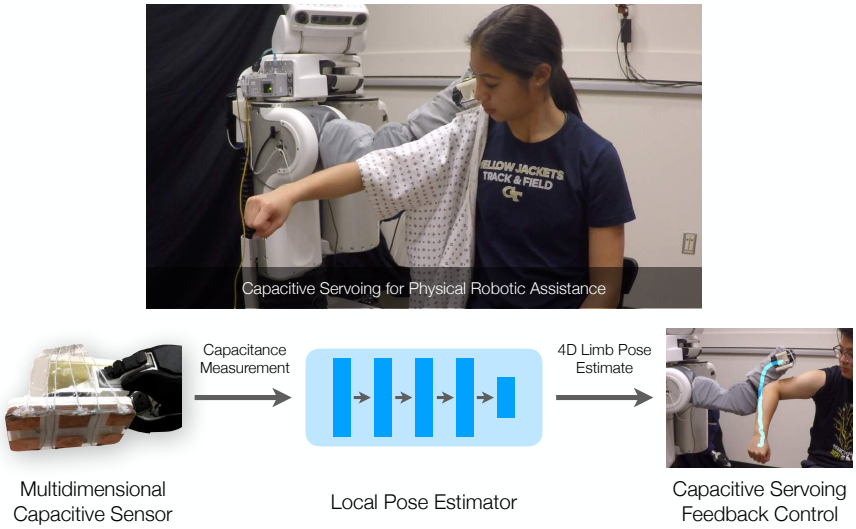


Figure 5.1: (Top) A PR2 robot uses capacitive servoing to track human motion while pulling a hospital gown onto a participant's arm. (Bottom) The capacitive servoing process. We use a multidimensional capacitive sensor in order to observe capacitance measurements during physical human-robot interaction. We feed these measurements to a trained pose estimation model which provides a relative 4D pose estimate for a nearby human limb. We then use a feedback controller in order for a robot's end effector to follow the contours along a human limb and adapt to human motion in real time.

own [76, 64]. However, accurately sensing the human body and tracking motion remains challenging for assistive robots, especially under the presence of visual occlusions.

In this chapter, we introduce and characterize a multidimensional capacitive servoing technique that enables a robot to track the local pose of a person's limb in real time during physical human-robot interaction (Figure 5.1). Capacitive servoing consists of a feedback control loop that leverages measurements from an array of conductive capacitive electrodes mounted to a robot's end effector. Each electrode generates an electric field, which in turn allows a robot to sense nearby conductive materials, such as a human body. A key benefit of capacitive sensing is the ability for robots to sense the human body through some opaque materials, including garments and wet cloth. Such materials often visually occlude a robot's sight of a person during caregiving tasks, such as dressing [9] and bathing.

For a robot to perform servoing around human limbs, we train a local pose estimation model that takes a time series of capacitance measurements from an array of six capacitive

electrodes and estimates a relative pose offset between a capacitive sensor mounted at the robot's end effector and a person's limb. This pose estimate includes the relative position, $\boldsymbol{p} = (p_y, p_z)$, as well as the pitch and yaw orientation, $\boldsymbol{\theta} = (\theta_y, \theta_z)$, between the capacitive sensor array and a nearby human limb. Yaw rotation is defined with respect to the central axis of the limb, whereas the relative position and pitch orientation are defined with respect to a coordinate frame at a nearby point on the surface of the limb. Using this representation, a robot is able to traverse its end effector along the upper surface of human limbs.

We then use these estimates with feedback control, which allows a robot to move its end effector along trajectories that follow the contours of a person's limbs. By conducting a study with 12 human participants, we investigate performance of this new servoing technique through several metrics, including task success rates, generalization, sensing ranges, and pose estimation error. Our results demonstrate that capacitive servoing can enable a robot to maintain its end effector on average within 1 cm from a target goal near a human limb, while simultaneously traversing the limb. Across various human limb poses and limb sizes, contact between the robot's end effector and a human limb occurred for only one participant and lasted for under 0.2 seconds.

The authors' prior work in [77] introduced a multidimensional capacitive sensor, which a robot used to provide dressing and bathing assistance to four able-bodied participants. Since this prior work was application-driven, it *did not* investigate the general capabilities and limitations of capacitive sensing for physical human-robot interaction. The current chapter significantly builds on work in [77] by characterizing a capacitive servoing control scheme for physical human-robot interaction and for controlling a robot's end effector to move along the contours of human limbs. In comparison to prior work, in this chapter we make the following contributions:

- We provide a formalized control scheme for capacitive servoing along with an intuition and design overview of multi-electrode capacitive sensors for sensing the pose of human limbs along multiple axes.

- We conducted a human-robot interaction study with 12 able-bodied participants where we investigated how capacitive servoing can enable a robot to control its end effector both proximally and distally along the contours of a human limb, which is the key component for performing several robotic caregiving tasks around the body.
- We characterize the sensing ranges of an end effector-mounted capacitive sensor and the generalization of human limb pose estimation across people with different body size.

5.3 Related Work

5.3.1 Capacitive Sensing

One of the most common applications for capacitive sensors are within consumer devices, such as touchscreens [78, 79], virtual reality controllers [80, 81], and wearables/clothing [82, 83]. Grosse-Puppendahl et al. [84] provide an overview of capacitive sensing within human-computer interaction, which includes a discussion of sensing modes, commonly used conductive materials, and capacitive sensing applications. Zimmerman et al. [85] also introduced several modes of operation and applications of capacitive sensing in human-computer interaction.

Within robotics, capacitive sensors have frequently been used as a form of proximity sensing [86]. Navarro et al. [87] provide a comprehensive overview of capacitive and proximity sensing in human-centered robotics, including applications, multimodal sensing techniques from literature that incorporate capacitance, and future challenges. Capacitive sensors have been mounted along the surface of robotic arms for avoiding collisions with people and stationary objects in the robot's environment [57, 56, 58, 88, 89, 90]. Non-contact capacitive sensing has also seen applications in controlling prostheses and exoskeletons [91, 92], tracking metallic objects near an industrial robot [93], teleoperation of mobile manipulators [94], and non-contact material recognition [59, 95]. When affixed

to the human body as wearable sensors, capacitive proximity sensors have also been used for limb motion recognition [96], activity recognition [97, 98], and health monitoring [99]. In comparison to prior capacitive sensing methods, we characterize a multidimensional capacitive sensor array for physical human-robot interaction, which enables a mobile manipulator to estimate relative human limb pose and perform servoing along the human body.

Capacitive sensors have also been widely used for contact-based tactile sensing with use cases in texture recognition [49] and for creating tactile artificial skins [51, 50], which can serve as a robotic skin for collision monitoring [54]. Capacitive sensors can be designed and integrated into artificial robotic skins to measure forces applied to the robot's body [52, 100, 55]. Capacitive tactile sensors have also been designed for robot fingertips and hands [53, 101, 102, 103, 104], which can provide high resolution force sensing during object grasping and dexterous manipulation. Capacitive sensing has been used in tandem with pneumatic sensing to estimate forces and deformation in a soft robotic finger [105]. A number of works have introduced and developed capacitive tactile proximity sensors, which are capable of both sensing conductive objects from a distance away and measuring applied contact forces [60, 65, 61, 62, 106].

In order to navigate a robot's end effector along the surface of stationary objects, Navarro et al. [62] introduced an arrangement of 2×2 capacitive sensors on the inside of a robot's parallel jaw gripper. The authors presented a closed-loop controller (proximity servoing) which operated on pairwise differences in sensor values. Using this setup, the robot would continually realign its end effector to the shape and orientation of an object, while it closed its end effector to grasp a flat plate or cylindrical rod. Navarro et al. [107] have also fitted a cylindrical robot end effector with an array of capacitive sensors for following the contours of stationary metallic objects. The authors estimate curvature by fitting circles to linear position estimates from the co-located capacitive sensors and then compute an angle between the capacitive sensors and the object curvature for proportional control. Similar to these two prior approaches, our work also uses capacitive sensing to define con-

trol trajectories for a robot's end effector. However, rather than sensing objects between a parallel jaw gripper or servoing around stationary metallic objects, we explore capacitive sensing for physical interaction with people, including following the contours along human limbs and tracking human limb motion. Additionally, we estimate the position and orientation offset between a robot's end effector and a human limb using a temporal data-driven approach on capacitance measurements.

5.3.2 Servoing

One of the most prominent servoing techniques is visual servoing [108, 109], which uses computer vision techniques to extract information from visual data as feedback to control a robot. Two common approaches are *image-based visual servoing* that uses features directly available in image data, and *position-based visual servoing*, which estimates the 3D pose of a robot relative to a calibrated camera pose for computing a kinematic error used in control. For a more in-depth overview on visual servoing methods, we refer the reader to [110, 111, 112, 113].

Towards robots that can interact with the human body, visual servoing has been applied in medical robotics contexts, such as tracking human body parts during surgery [114, 115] and commanding robotic surgical tools [116, 117, 118]. Visual servoing has also been used for collision avoidance with a human collaborator wearing motion capture equipment [119], to track human trajectories during human-robot collaboration [120], and for human-robot object transfers [121].

Beyond visual servoing, other modalities such as force or capacitive sensing have also been used in the control loop. Reichl et al. [122] present a servoing approach for tracking medical instruments within the human body by using electromagnetic sensors that do not require line of sight. Lepora et al. [123] use tactile servoing to navigate along edges of planar objects. Sutanto et al. [124] learn a dynamics model from human demonstrations for tactile servoing with a BioTac tactile sensing finger. Ding et al. [125, 89] attach time-

of-flight (ToF) and capacitive proximity sensor cuffs to a robot arm for active collision avoidance during robot motion while following a task trajectory. In comparison to other servoing methods, in this chapter we characterize a multidimensional capacitive servoing technique that enables a robot to navigate its end effector along the contours of human limbs, regardless of visual occlusions and without requiring contact with a person's body.

5.3.3 Human Pose Estimation for Physical Assistance

A number of approaches have been proposed for sensing human pose during physical human-robot interaction. One common pose estimation method during physical interaction is using visual features from an RGB or depth camera. For example, several works have used depth sensing for estimating human pose [10, 8, 68, 126] and tracking cloth features [28, 127, 128] while a robot helps a person or mannequin dress on a clothing garment. Jiménez et al. [129] provide an overview of various perception techniques for tracking cloth during assistive robotic tasks. Zlatintsi et al. [74, 130] present a bathroom setup with three Microsoft Kinects to capture RGB and depth information for human gesture and pose recognition in a robot-assisted bathing context. However, the human body can often become visually occluded during physical human-robot interaction tasks, where the robot's own arms may intersect line of sight of a person, or clothing can occlude the body during dressing assistance.

In healthcare and assistive robotics settings, sensing approaches that rely on force sensors [11], pressure mats [131, 67, 132], capacitive sensors [66, 133, 134], and feature extraction from occluded depth images [135, 68] have been applied to recover visually occluded human pose.

As an alternative to visual features, haptic measurements have been used to detect errors in robot-assisted dressing tasks when cloth gets caught on the human body [42, 47]. Haptic and kinematic measurements from a robot's end effector have been used during robot-assisted dressing to estimate the forces a fabric garment applied onto a person's sta-

tionary body [30, 70]. Several works have also combined force measurements with data from a depth camera to optimize personalized robotic dressing trajectories for a sleeveless jacket [11, 69, 136]. In contrast, capacitive sensing has the advantage of tracking human pose without applying force to a person's body and can directly sense the human body through many visually opaque materials, including cloth.

5.4 Sensor, Model, and Controller

In this section, we describe the capacitive servoing technique, including the multidimensional capacitive sensor and data-driven human pose estimation models. Along with this, we provide the process for collecting capacitance data for pose estimation, and the capacitive servoing feedback control loop for controlling a robot's end effector to move along the contours of a human limb.

5.4.1 Multidimensional Capacitive Sensor Design

Our capacitive sensor design consists of six positively charged capacitive electrodes, arranged in a 3-by-2 grid and adhered to a polycarbonate plate (Figure 5.2). The entire sensor with the polycarbonate mounting plate is a 11.5 cm $\times$ 8.5 cm $\times$ 1 mm rectangular surface with rounded corners. Each electrode is a 3 cm $\times$ 3 cm square of copper foil, which we connected to a commercially available Teensy 3.2 microcontroller board. The Teensy has dedicated hardware components for capacitive proximity sensing, including an analog voltage comparator and a 1 pF internal reference capacitor. With this setup, we can sample capacitance measurements at frequencies over 100 Hz from all six electrodes simultaneously. We then attached this capacitive sensor to the bottom of an assistive tool held by a robot's end effector, shown in Figure 5.2. This assistive tool enables a robot to more easily hold task relevant objects, such as fabric garments or washcloths.

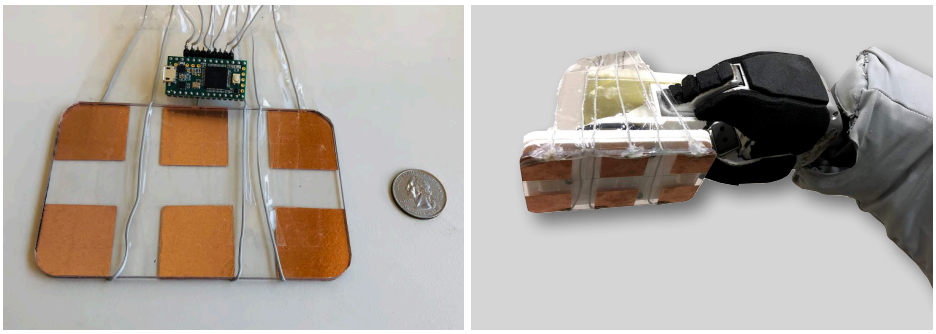
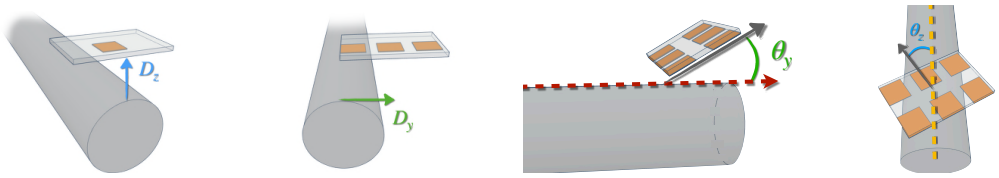


Figure 5.2: (Left) Six-electrode capacitive sensor connected to a Teensy microcontroller board, with a U.S. quarter shown for sizing. (Right) Capacitive sensor mounted to the bottom of an assistive tool, which allows a robot to more easily hold task relevant objects (garments and washcloths).

5.4.2 Multidimensional Capacitive Sensing Intuition

Capacitive sensing is frequently used as a means to sense a nearby human body for human-computer interaction [84] and some human-robot interaction contexts [61, 58]. A simple *self-capacitance* capacitive sensor consists of a conductive electrode connected to a power source, which in turn generates an electric field. An external conductive object, such as a human limb, that intersects this electric field causes an increase in capacitance that can be measured from the electrode. Capacitance, or the change in electric charge stored on the electrode (given a fixed electric potential), is then inversely proportional to the distance between the electrode and the human limb.

To gain an intuition for this multidimensional capacitive sensor design for sensing human limb pose, we can begin by representing a human limb as an abstract cylinder, as shown in Figure 5.3 and Figure 5.4. As demonstrated in [66, 133], only a single electrode is needed to determine the vertical distance between the sensor and the surface of a human limb, along the Z-axis. This scenario is depicted in Figure 5.3 (a) with a single capacitive electrode vertically above the abstract cylinder. By adding two additional electrodes in parallel, the sensor can also measure lateral displacement along the Y-axis, as displayed in Figure 5.3 (b). Finally, by adding a second row of three electrodes, this six-electrode capacitive sensor can measure pitch rotation to the surface of the limb (θ_y, Figure 5.3 (c))



(a) (b) (c) (d)

Figure 5.3: (a) Sensing vertical displacement D_z between the capacitive sensor and surface of a cylinder is feasible with only a single electrode. (b) A row of three capacitive electrode enables sensing both lateral displacement D_y and vertical displacement. (c), (d) Adding a second row of electrodes allows for sensing pitch rotation θ_y to the surface of the cylinder and yaw rotation θ_z with respect to the central axis.

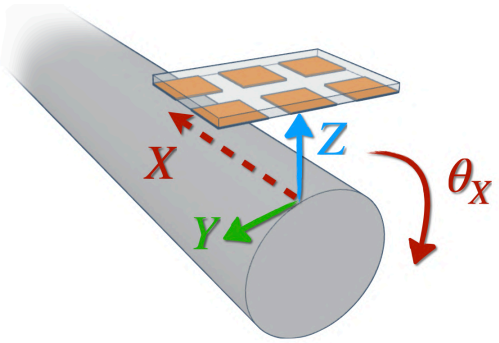


Figure 5.4: A multidimensional capacitive sensor above an infinitely long cylinder, which is an abstract representation of a human limb. We sense the relative translation and orientation between the capacitive sensor array and the origin of a coordinate frame at a point on the surface of the cylinder (limb). Note that sensing translation and rotation with respect to the X-axis along a cylinder (X and θ_X shown in red) is infeasible.

and yaw rotation with respect to the central axis of the limb (θ_z, Figure 5.3 (d)).

Overall, we can use this 3×2 grid of electrodes to sense the 4-dimensional relative pose of a human limb (or conductive cylindrical object) with respect to the sensor. This pose includes the 2-dimensional translation and 2-dimensional rotation between the capacitive sensor array and a nearby limb. The 2-dimensional translation (D_y, D_z) and pitch rotation (θ_y) are defined with respect to a coordinate frame at a point on the surface of the limb. We define this point as the highest vertical point on a cross section of the limb—a cross section that is perpendicular to the limb's central axis and intersects the center of the capacitive sensor (see Figure 5.4). Yaw rotation (θ_z) is then defined with respect to the central axis of the limb. Note that we do not sense translation or rotation (roll, θ_x) around the X-axis of

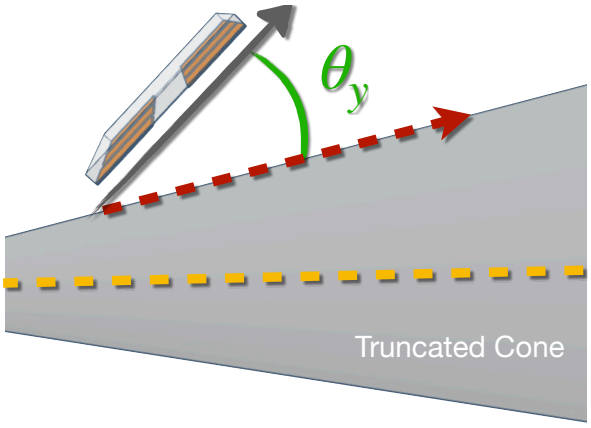


Figure 5.5: θ_y is represented geometrically as the orientation between the sensor and the local limb surface (red dashed line) rather than to the limb's central axis (yellow dashed line). This representation allows capacitive servoing to generalize to non-cylindrical limbs which can be more accurately modeled as a truncated cone (conical frustum).

the limb due to a cylinder's self-similarity along its central axis, as depicted in Figure 5.4. Figure 5.5 highlights the importance of geometrically modeling pitch rotation to the surface of a limb rather than the central axis. Doing so allows capacitive servoing to generalize to non-cylindrical limbs that are instead more accurately modeled as a truncated cone (conical frustum).

Figure 5.6 depicts an example of how capacitance measurements from this six-electrode sensor vary over time as a PR2 robot moves the sensor over a human arm. The sensor remains 5 cm vertically above the arm as it translates laterally across the arm. Since the sensor begins on the left side of the arm, the right column of capacitive electrodes initially measure a high capacitance. Once the sensor is centered above the arm (around the 1.5 second mark), the center row of electrodes record a high capacitance, while the four corner electrodes record lower capacitance readings. This trend continues as the sensor translates to the right side of the arm at which point the left column of electrodes measure the greatest capacitance.

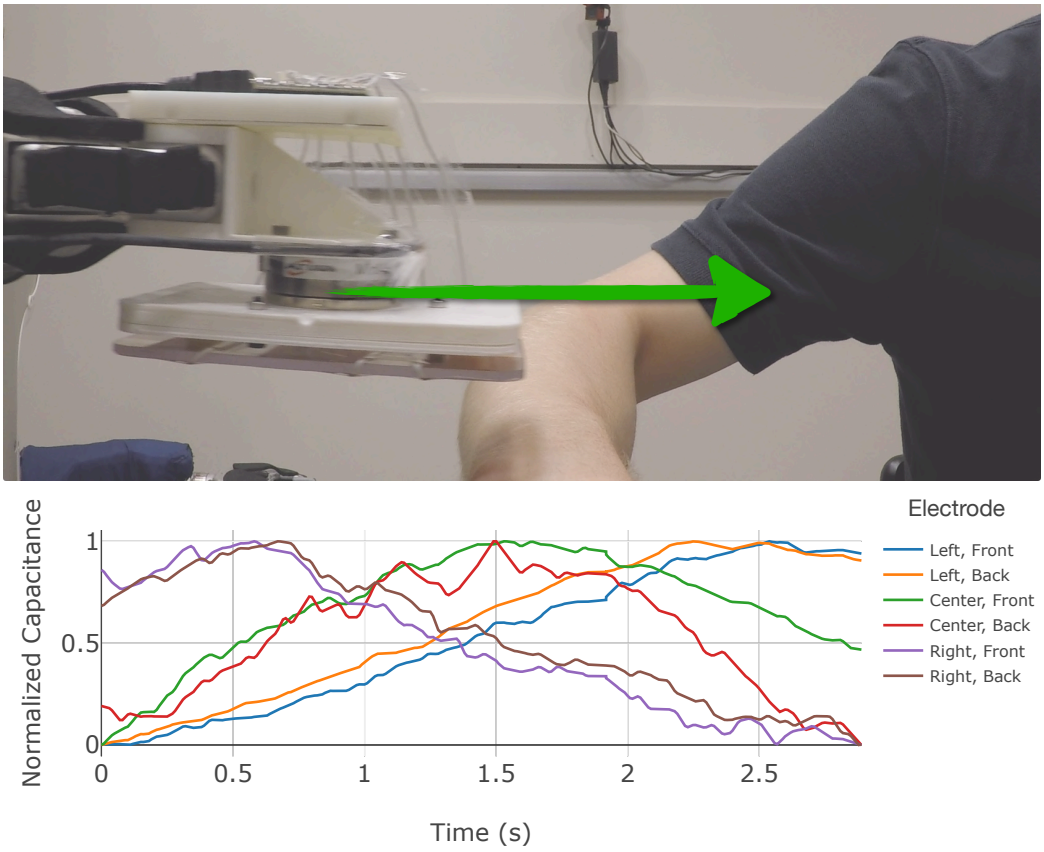


Figure 5.6: Capacitance measurements from all six capacitive electrodes as the robot moves the sensor from one side of a person's arm to the other. The sensor remains ∼5 cm vertically above a person's arm. We apply a finite impulse response (FIR) filter with numerator and denominator coefficients of 1 and 0.02, respectively. We normalize all signals to the range of [0, 1].

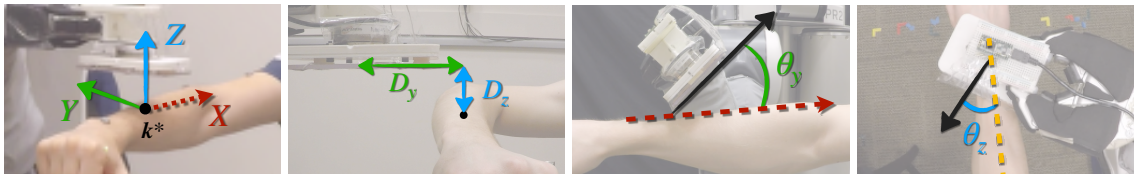
5.4.3 Estimating Relative Pose of a Human Limb

Given measurements from this robot-mounted six-electrode capacitive sensor, we then consider how to estimate the geometric transpose between this capacitive sensor and a nearby human limb.

Using a single electrode capacitive sensor, [66] and [133] demonstrated how non-linear least squares optimization can be used to fit a model for estimating the one-dimensional vertical distance between a robot's end effector and a human arm. Inspired by the classical capacitance equation for a parallel plate capacitor, this prior one-dimensional model, $d(C) = \frac{\alpha}{C+\beta}$, estimates the shortest distance between a capacitive sensor and a human limb given a single capacitance measurement, C, and optimized constants, $\{\alpha, \beta\}$. Yet, this straightforward model faces several challenges with respect to estimating both position and orientation of a human limb. The model is designed around a single capacitance measurement and lacks an evident generalization to multiple sensory inputs and pose outputs. In addition, this one-dimensional approach does not model variable electromagnetic interference (EMI) or crosstalk between adjacent electrodes, which are present in multi-electrode sensor configurations.

To overcome these challenges, we consider a model $f : \mathbb{R}^{6h} \rightarrow \mathbb{R}^4$, that maps a temporal window of prior capacitance measurements over h previous time steps from all six electrodes to a 4-dimensional relative pose estimate of a human limb. This model, $f(\boldsymbol{c}_{t-h+1:t})$, outputs an estimated relative pose $\hat{\boldsymbol{p}}_t$ of a nearby human limb given a window of prior capacitance measurements, $\boldsymbol{c}_{t-h+1:t}$, from time step $t - h + 1$ to the current time step t. Each capacitance measurement $\boldsymbol{c}_t = (c_1, \ldots, c_6)$ is a vector of measurements from all six capacitive electrodes.

The pose estimate $\hat{\boldsymbol{p}}_t = (\hat{D}_{t,y}, \hat{D}_{t,z}, \hat{\theta}_{t,y}, \hat{\theta}_{t,z})$ includes the relative position $\boldsymbol{D}_t = (D_{t,y}, D_{t,z})$ and orientation of the limb $\boldsymbol{\theta}_t = (\theta_{t,y}, \theta_{t,z})$, as shown in Figure 5.7. The relative 2-dimensional translations are represented in relation to a point on the surface of a human limb. Consider a vector $\boldsymbol{n}$ along the central axis of a human limb that is near the capacitive



(a) (b) (c) (d)

Figure 5.7: (a) The local coordinate frame at the top most vertical point $\boldsymbol{k}^*$ along the nearest perpendicular cross section of a human limb, as defined in (5.1). (b) The lateral D_y and vertical D_z displacement between the center of the capacitive sensor array and the point on the limb. (c) Pitch rotation θ_y between the capacitive sensing plane and the local limb surface. (d) Relative yaw rotation θ_z difference between the capacitive sensing array and the central axis of a human limb.

sensor array and consider a point along the central axis $\boldsymbol{k}_0$ which is closest to the center of the capacitive sensor array. We define a set of all points $\boldsymbol{k}$ such that $\boldsymbol{n} \cdot (\boldsymbol{k} - \boldsymbol{k}_0) = 0$ forms a plane perpendicular to the central axis and a cross section of the limb. 2-dimensional translations $\boldsymbol{D}_t$ between the capacitive sensor and human limb are then defined with respect to the top most point in this plane along the surface of the limb,

$$\begin{aligned} \boldsymbol{k}^* &= \arg\max_{\boldsymbol{k} \in \mathcal{H}} \quad \boldsymbol{k} \cdot \boldsymbol{e}_z \\ \text{subject to} \quad & \boldsymbol{n} \cdot (\boldsymbol{k} - \boldsymbol{k}_0) = 0, \end{aligned} \tag{5.1}$$

where $\boldsymbol{e}_z = (0, 0, 1)^T$ is a constant vector and $\mathcal{H}$ is the surface of the human limb.

Figure 5.7 (a) visually depicts this point $\boldsymbol{k}^*$ along a human arm near the capacitive sensor array. Pitch orientation $\theta_{t,y}$ is defined according to the Y-axis of a coordinate frame at the point $\boldsymbol{k}^*$ on this surface of the limb, as shown in Figure 5.7 (a) and (c), whereas yaw orientation $\theta_{t,z}$ is defined with respect to the central axis of the limb, depicted in Figure 5.7 (d).

As discussed in Chapter 5.4.2, this sensing approach operates with respect to the Y- and Z-axes of a human limb, as it is difficult to sense relative position and orientation changes around the X-axis along the surface of the limb, due to the approximate cylindrical shape of a limb (Figure 5.7 (a)).

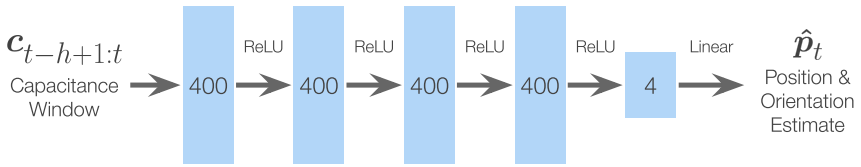


Figure 5.8: The human pose estimation model used in capacitive servoing, implemented as a fully-connected neural network. This model estimates the current relative human limb pose $\hat{\boldsymbol{p}}_t$ given a window of measurements $\boldsymbol{c}_{t-h+1:t}$ from a capacitive sensor.

To estimate the relative pose of a human limb near the capacitive sensor, we leveraged a data-driven implementation of $f(\boldsymbol{c}_{t-h+1:t})$ using a fully-connected neural network architecture, as depicted in Figure 5.8. The network has four 400 node layers with a ReLU activation after each layer and a final output layer of four nodes with a linear activation. This final layer outputs an estimate, $\hat{\boldsymbol{p}}_{\boldsymbol{t}}$, of the relative pose of a person's limb with respect to the capacitive sensor. We trained this model over 100 epochs using the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$, a batch size of 128, and a learning rate of 0.001.

At any given time step t, our implementation of this model takes as input a window of the 50 most recent measurements from the six capacitive sensor electrodes, $\boldsymbol{c}_{t-49:t} \in \mathbb{R}^{50\times 6}$. When measured at a frequency of 100 Hz, this results in a 0.5 second window of prior capacitance data. We then vectorize these measurements for the network, resulting in a model of the form $f : \mathbb{R}^{300} \rightarrow \mathbb{R}^4$. By using a window over prior measurements, this model is able to learn the temporal aspects of how capacitance measurements vary over time as the sensor translates and rotates around human limbs. This approach also provides an opportunity to implicitly model and account for crosstalk (electromagnetic interference) between the co-located sensors during human-robot interaction.

5.4.4 Data Collection

We trained our human pose estimation model for capacitive servoing using approximately 630,000 data pairs $(\boldsymbol{c}_{t-h+1:t}, \boldsymbol{p}_t)$ of time-varying capacitance measurements and relative limb poses. These data pairs were captured as a PR2 robot moved the capacitive sensor

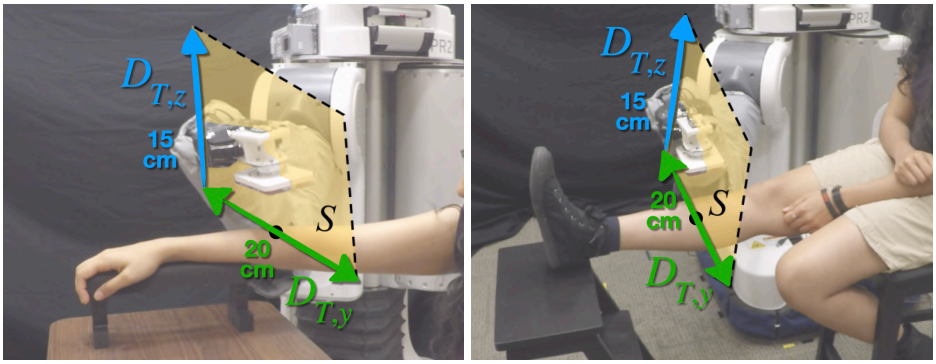


Figure 5.9: During data collection, a participant elevated their arm and leg at a stationary pose parallel to the ground using an armrest and footstool. The capacitive sensor then started at a relative pose of $\boldsymbol{p}_0 = (0, 0, 0, 0)$ above the limb. The yellow highlighted region represents the space of target end effector positions $(D_{T,y}, D_{T,z})$ relative to the point $\boldsymbol{k}^*$ on the surface of the limb.

around a human participant's stationary arm and leg. Data Collection 1 in the Appendix outlines the capacitance data collection process between the robot and a human participant.

As shown in Figure 5.9, the participant began with their arm or leg elevated outward parallel to the ground with the support of an armrest and footstool. We position the robot's end effector and capacitive sensor above the participant's limb with an initial relative pose of $\boldsymbol{p}_0 = (0, 0, 0, 0)$. The robot then selects a target end effector pose above the limb $\boldsymbol{p}_T$ from a uniform distribution, along with translational and rotational velocities for its end effector to move towards the target. The bounded space S of end effector poses above a person's static limb is depicted in Figure 5.9. At each time step t while moving its end effector to the target, the robot records capacitance measurements $\boldsymbol{c}_t$ from the six capacitive sensor electrodes coupled with the current pose of the end effector $\boldsymbol{p}_t$ from forward kinematics. Once the robot reaches the target state $\boldsymbol{p}_t \approx \boldsymbol{p}_T$, the robot iteratively selects and moves towards a new target pose $\boldsymbol{p}_T$ for a total of $N = 500$ end effector trajectories above the limb. The detailed data collection process is provided in the Appendix.

We repeat the data collection process for three locations along the arm (wrist, forearm, upper arm) and three locations along the leg (ankle, shin, knee). Conducting $N = 500$ iterations resulted in $\sim$17.5 minutes of data collection for each location along the arm and

leg. We recorded measurements at a frequency of 100 Hz ($\tau = 100$) for a total of 630,000 samples $(\boldsymbol{c}_t, \boldsymbol{p}_t)$. We then windowed all of the observed data into pairs $(\boldsymbol{c}_{t-h+1:t}, \boldsymbol{p}_t)$ for model training and evaluation. We collected these measurements with a single human participant, and in Chapter 5.5 we demonstrate how a trained pose estimation model can generalize to multiple participants.

5.4.5 Capacitive Servoing Feedback Control

At any given time step t when a robot is operating near the human body (e.g. during the evaluation Chapter 5.5), the robot observes a window of the most recent capacitance measurements $\boldsymbol{c}_{t-h+1:t}$ from the capacitive sensor. As discussed in Chapter 5.4.3, we can feed these measurements through our trained pose estimation model $f(\boldsymbol{c}_{t-h+1:t})$, which produces a relative pose estimate $\hat{\boldsymbol{p}}_t$ of the nearby person's limb. In this section, we discuss the general control scheme used in capacitive servoing to adapt to human motion and follow the contours of a human limb. In addition, we detail our specific implementation of this controller, which our robot used to interact with human participants.

Algorithm 4 outlines the capacitive servoing control scheme for robots, which uses a high-level Cartesian controller to define end effector trajectories. Before control begins, the robot fills the temporal window of capacitance measurements used for pose estimation (lines 2-3). At each time step during execution, the robot observes the current capacitance readings from all capacitive electrodes, $\boldsymbol{c}_t$ (line 6). Just as during data collection (Chapter 5.4.4), with our implementation the robot collects these measurements at a frequency of 100 Hz ($\tau_d = 100$).

In order to update its end effector trajectory, the robot begins by computing the current tracking error $\boldsymbol{e}(t)$ (line 8). This error is calculated as the difference between the desired pose offset $\boldsymbol{p}_{desired} = (D_y, D_z, \theta_y, \theta_z)$ and the estimated relative pose $\hat{\boldsymbol{p}}_t$ from our trained model $f(\boldsymbol{c}_{t-h+1:t})$.

The robot then computes a proposed action $\boldsymbol{u}(t) = (u_y(t), u_z(t), u_{\theta_y}(t), u_{\theta_z}(t))$ (line 9)

Algorithm 4 Capacitive Servoing for Human Interaction

1: **Given:** f: pose estimation model,
u: control policy,
$\boldsymbol{p}_{desired}$: target end effector pose,
h: capacitance window size,
v_x: forward velocity,
τ_d: data capture frequency,
τ_u: control frequency.
2: **for** $t = 1, \ldots, h-1$ **do**
3: $\quad \boldsymbol{c}_t \leftarrow$ GetCapacitanceMeasurements().
4: $t \leftarrow h$.
5: **while** *task not completed* **do**
6: $\quad \boldsymbol{c}_t \leftarrow$ GetCapacitanceMeasurements().
7: $\quad$ **if** $t \bmod \lfloor \frac{\tau_d}{\tau_u} \rfloor = 0$ **then**
8: $\quad\quad \boldsymbol{e}(t) \leftarrow \boldsymbol{p}_{desired} - f(\boldsymbol{c}_{t-h+1:t})$.
9: $\quad\quad$ Compute $(u_y(t), u_z(t), u_{\theta_y}(t), u_{\theta_z}(t))$ using (5.2).
10: $\quad\quad \boldsymbol{u}_{t,x} \leftarrow$ TransformToRobotFrame($(\frac{v_x}{\tau_u}, 0, 0)$).
11: $\quad\quad \boldsymbol{\omega}_t, \boldsymbol{\phi}_t \leftarrow$ GetCurrentEndEffectorPose().
12: $\quad\quad \boldsymbol{\omega}_t \leftarrow \boldsymbol{\omega}_t + (0, u_y(t), u_z(t)) + \boldsymbol{u}_{t,x}$.
13: $\quad\quad \boldsymbol{\phi}_t \leftarrow \boldsymbol{\phi}_t + (0, u_{\theta_y}(t), u_{\theta_z}(t))$.
14: $\quad\quad \boldsymbol{\alpha}_t \leftarrow$ InverseKinematics($\boldsymbol{\omega}_t, \boldsymbol{\phi}_t$).
15: $\quad\quad$ SendToActuators($\boldsymbol{\alpha}_t$).
16: $\quad t \leftarrow t + 1$.

towards the target pose $\boldsymbol{p}_{desired}$. This action consists of both translation $(u_y(t), u_z(t))$ and orientation $(u_{\theta_y}(t), u_{\theta_z}(t))$ steps for the pose of the end effector. The robot also moves along the X-axis of its end effector at a fixed velocity v_x (see Figure 5.10), which ensures that the end effector traverses either proximally or distally along a person's limb. To do so, the robot transforms this fixed velocity action (line 10) from the end effector coordinate frame to its central base coordinate frame. To execute this entire action, we form a new target end effector pose for the robot by adding the action to the current end effector pose $\boldsymbol{\omega}_t, \boldsymbol{\phi}_t$ and then use inverse kinematics to compute target angles for each robot actuator (lines 11-15).

In our implementation, we define a PD (proportional-derivative) feedback controller to

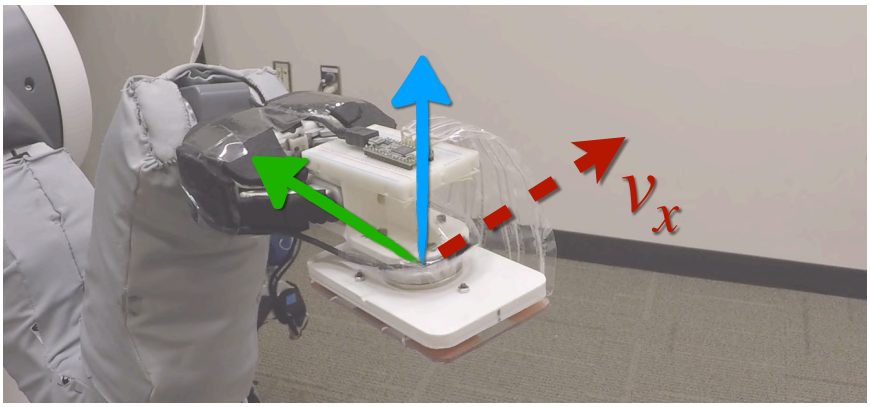


Figure 5.10: In addition to the control action $u(t)$ during capacitive servoing, the robot moves along the X-axis of its end effector at a velocity of v_x. Doing so ensures the robot navigates along the surface of a human limb during physical interaction.

compute actions $\boldsymbol{u}(t)$ for the robot's end effector. We define this controller as,

$$\boldsymbol{u}(t) = \boldsymbol{K}_p \boldsymbol{e}(t) + \boldsymbol{K}_d \dot{\boldsymbol{e}}(t). \tag{5.2}$$

Here, $\boldsymbol{K}_p = \text{diag}(0.025, 0.025, 0.1, 0.1)$ and $\boldsymbol{K}_d = \text{diag}(0.0125, 0.0125, 0.025, 0.025)$ represent diagonal matrices for the proportional and derivative gains, which we tuned to generate smooth end effector motion when tracking the contours of a person's limb. We also instruct the robot to move 2 cm/s along the X-axis of its end effector, i.e. $v_x = 2$. In this work, we emphasize that capacitive servoing can be a successful control technique around the human body, even with straightforward feedback controllers, such as PD control.

In comparison to the measurement capture frequency, which we run at 100 Hz, this control sequence (lines 8-15) can be run at lower frequencies (e.g. at 10 Hz with $\tau_u = 10$). Doing so can be useful in cases where performing a forward pass through the pose estimation model f is slower than the data capture frequency. The robot performs this control loop (line 5) continuously until a given task is complete. For the tasks explored in this chapter, capacitive servoing continues until the end effector has traversed a person's entire limb of length l (as determined by end effector velocity along the X-axis v_x), such as navigating from the person's hand up to their shoulder. Additionally, we implement a force

threshold monitor using the 6-axis ATI force/torque sensor mounted on the assistive tool, which halts the robot's movements if the robot makes contact with a person and applies a magnitude of force greater than 10 N.

We control 4 degrees of freedom for the end effector along the Y- and Z-axes $(D_{t,y}, D_{t,z}, \theta_{t,y}, \theta_{t,z})$ and we control translation along the X-axis via v_x. We do not control or constrain rotation around the X-axis, which provides the robot and inverse kinematics some flexibility as it rotates around sharply bent limb joints, as depicted in Chapter 5.5.3. In practice, we did not observe significant drift in end effector rotation along the X-axis $\theta_{t,x}$, except in cases when the human limb was farther away from the sensor than the sensor's functional 15 cm sensing range (sensing ranges are characterized in Chapter 5.5.1).

With this controller setup, $\boldsymbol{p}_{desired}$ can be dynamically changed to accommodate the robot's current task. In our evaluations with human participants (Chapter 5.5), we set $\boldsymbol{p}_{desired} = (0, 5, 0, 0)$ so that the robot's end effector stays 5 cm above a participant's limb and maintains the same orientation as the limb. For other tasks, such as providing bathing assistance (see Chapter 5.5.4), we can set $\boldsymbol{p}_{desired} = (0, 1, 0, 0)$ so that a washcloth attached to the capacitive sensor makes direct contact with a person's skin.

5.5 Capacitive Servoing Characterization

In order to characterize the proposed capacitive servoing technique, including the trained pose estimation model and capacitive sensor limits, we conducted a human-robot study with 12 able-bodied human participants (6 females and 6 males).

We conducted several experiments with each participant to evaluate sensing ranges and generalization of capacitance measurements across people. As part of our experiments, we evaluated how a PR2 robot can use multidimensional capacitive servoing to move its end effector along human limbs—a task which is valuable for robotic caregiving and several physical human-robot interaction contexts. During these studies, the robot used the six-electrode capacitive sensor array described in Chapter 5.4.

We obtained informed consent from all participants and approval from the Georgia Institute of Technology Institutional Review Board (IRB). We recruited participants to meet the following inclusion/exclusion criteria: $\geq$ 18 years of age; fluent in written and spoken English; and have not been diagnosed with ALS or other forms of motor impairments. Participant ages ranged from 18 to 27 years old. Participant arm lengths (measured from wrist to shoulder) varied between 56 cm to 64 cm and leg lengths (measured from ankle to hipbone) varied between 75 cm to 97 cm. Video sequences of experiments and results can be found in the supplementary video.

5.5.1 Characterizing Sensing Ranges

We began by characterizing the effective sensing range of the six-electrode capacitive sensor and trained pose estimator. We did this by computing pose estimation error as the PR2 moved the capacitive sensor away from each participant's stationary arm.

We first investigated the translational sensing range of the capacitive sensor along the two relative position axes (D_y, D_z). The PR2 started by holding the capacitive sensor centered 3 cm above each participant's forearm. The robot's end effector then followed a linear trajectory in a random direction away from a participant's forearm within the Y-Z plane until the capacitive sensor was 20 cm away from the forearm. This resulted in the semicircle region of end effector positions shown in Figure 5.11. Throughout a linear trajectory, the robot recorded capacitance measurements at 100 Hz as the end effector maintained a velocity uniformly selected from [3 cm/s, 10 cm/s] and a relative orientation parallel to the participant's arm ($\theta_y = 0$ and $\theta_z = 0$). We use variable velocities to account for noise in actuation, variability in feedback control, and human motion, all which can occur in typical operation. Once a trajectory completed, the end effector returned to the initial starting point above a participant's forearm and repeated the process. In total, the robot executed 20 trajectories above each of the 12 participant's forearms, resulting in 93,000 data pairs $(\boldsymbol{c}_t, \boldsymbol{p}_t)$ across 240 linear trajectories. We preprocessed these data into temporal windows of 50 ca-

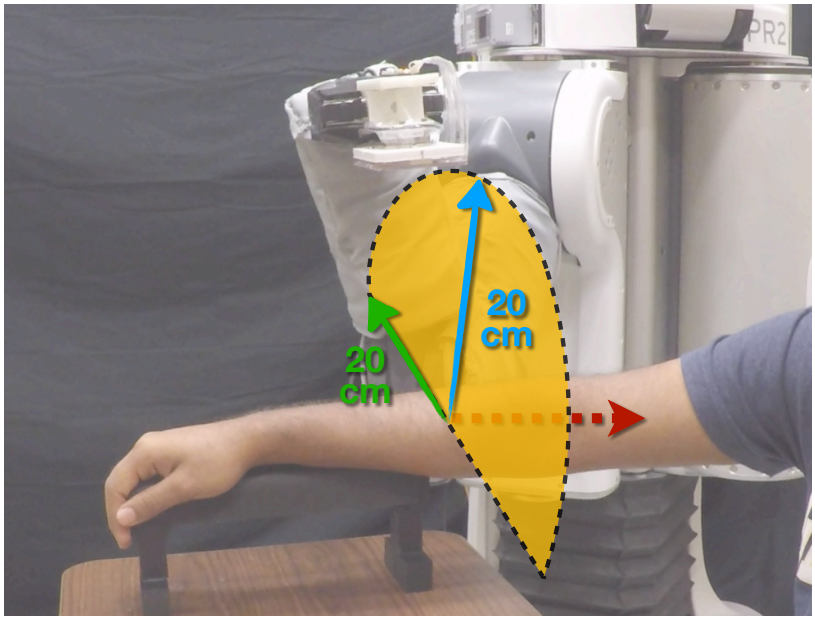


Figure 5.11: To evaluate the sensing ranges of the implemented six-electrode capacitive sensor, the PR2 recorded data pairs $(\boldsymbol{c}_t, \boldsymbol{p}_t)$ while following 20 linear trajectories above each participant's forearm. The robot would linearly translate its end effector in a random direction within the Y-Z plane until the capacitive sensor was 20 cm from the forearm. This resulted in the semicircle region of target end effector positions shown in yellow.

pacitance measurements $\boldsymbol{c}_{t-49:t}$, as discussed in Chapter 5.4.3, and computed relative pose estimates using our trained model $f(\boldsymbol{c}_{t-49:t}) = \hat{\boldsymbol{p}}_t = (\hat{D}_{t,y}, \hat{D}_{t,z}, \hat{\theta}_{t,y}, \hat{\theta}_{t,z})$.

These linear trajectories away from a limb intentionally differ slightly from the random end effector trajectories found in the training data distribution. These linear trajectories are closer aligned to the trajectories we expect our feedback controller to exhibit during typical operation and include end effector poses that extrapolate outside of the training distribution. In the following section we also discuss repeating the data collection process with all 12 participants and evaluating generalization across the resulting random end effector trajectories.

We quantify position estimation errors by computing the mean absolute error $\overline{\boldsymbol{D}_\epsilon} = \frac{1}{2}||\boldsymbol{D}_t - \hat{\boldsymbol{D}}_t||_1$ between the ground truth relative position $\boldsymbol{D}_t = (D_{t,y}, D_{t,z})$ and estimated position $\hat{\boldsymbol{D}}_t$ of a participant's forearm at time step t. Figure 5.12 depicts the mean position estimation error $\overline{\boldsymbol{D}_\epsilon}$ at varying end effector locations away from a human forearm. The X-

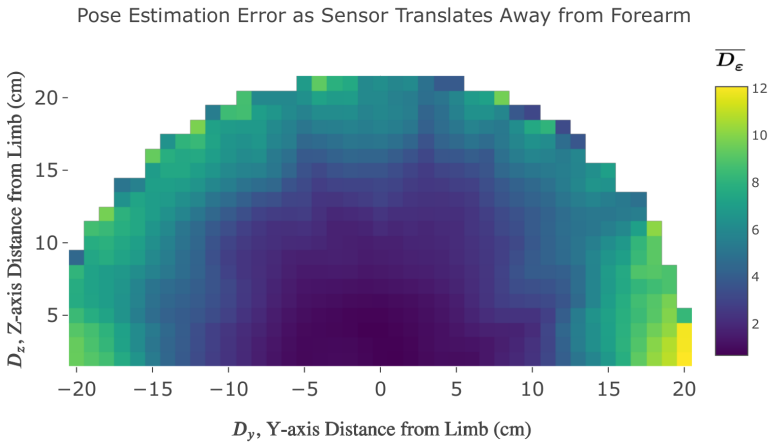


Figure 5.12: Mean estimation error $\overline{\boldsymbol{D}_\epsilon}$ (in centimeters) of the trained pose estimation model when the capacitive sensor was at various positions away from a human forearm. The X- and Y-axis show the ground truth lateral position $\boldsymbol{D}_y$ and vertical position $\boldsymbol{D}_z$ of the capacitive sensor away from the stationary human forearm, respectively. Each 1 cm $\times$ 1 cm grid cell with center (y, z) is colored to represent the mean magnitude of pose estimation error present over all data pairs $(\boldsymbol{c}_{t-49:t}, \boldsymbol{p}_t)$ near the cell where $D_y \in [y - 1.5, y + 1.5]$ and $D_z \in [z - 1.5, z + 1.5]$.

axis represents the ground truth relative lateral position $\boldsymbol{D}_y$ between the capacitive sensor and human forearm, whereas the Y-axis enumerates the relative vertical position $\boldsymbol{D}_z$. For each 1 cm $\times$ 1 cm grid cell with center (y, z) in Figure 5.12, we computed the mean error over all data pairs $(\boldsymbol{c}_{t-49:t}, \boldsymbol{p}_t)$ that lie near the cell, i.e. $D_y \in [y - 1.5, y + 1.5]$ and $D_z \in [z - 1.5, z + 1.5]$.

When the center of the capacitive sensor array was within a 10 cm radius from a participant's forearm, the mean position estimation error $\overline{\boldsymbol{D}_\epsilon}$ was less than 1.1 cm—shown as the center dark blue region in Figure 5.12. Between a 10 cm to 15 cm radius from the arm, error increased to 2.8 cm. When the capacitive sensor was greater than 15 cm from the forearm, the mean position error from our pose estimation model was 5.6 cm, which we see by the green shading along the outer semicircle region in Figure 5.12. From this, we can postulate that this capacitive sensing technique and pose estimation model has a functional sensing range of up to 15 cm from a human arm. In Chapters 5.5.3 and 5.5.4, we demonstrate that this sensing range is enough for a mobile manipulator to accurately follow

the contours of a person's arm and assist participants with two activities of daily living.

We also quantify the rotational sensing range of the capacitive sensor along the two relative orientation axes (θ_y, θ_z). The PR2 initialized its end effector above each participant's forearm such that the capacitive sensor started 5 cm above the forearm ($D_y = 0$ cm, $D_z = 5$ cm) with an initial orientation parallel to the arm ($\theta_y = 0$, $\theta_z = 0$). The robot then rotated its end effector to a random target orientation around the Y- and Z-axes until the capacitive sensor was oriented 45 degrees away from its starting orientation (while remaining 5 cm above the forearm), i.e. $||\boldsymbol{\theta}||_2 = 45$ degrees. We again repeated this process 20 times for each participant, for a total of 86,000 data pairs $(\boldsymbol{c}_{t-49:t}, \boldsymbol{p}_t)$ and associated pose estimates $f(\boldsymbol{c}_{t-49:t})$ across 240 rotation trajectories. The supplemental video visually depicts these trajectories above a participant's forearm.

We quantify orientation estimation errors via the mean absolute error $\overline{\boldsymbol{\theta}_\varepsilon} = \frac{1}{2}||\boldsymbol{\theta}_t - \hat{\boldsymbol{\theta}}_t||_1$ between the true relative orientation $\boldsymbol{\theta}_t = (\theta_{t,y}, \theta_{t,z})$ and estimated orientation $\hat{\boldsymbol{\theta}}_t$ of a participant's forearm. Figure 5.13 visualizes the mean orientation estimation error $\overline{\boldsymbol{\theta}_\varepsilon}$ as the capacitive sensor rotates above a human forearm in different directions. For all points in which the capacitive sensor had a less than 30 degree orientation from its starting orientation above a participant's forearm ($||\boldsymbol{\theta}_t||_2 < 30$ degrees), the mean orientation error was $\overline{\boldsymbol{\theta}_\varepsilon} = 6.2$ degrees, depicted as the blue center region in Figure 5.13. Yet, when the sensor was between 30 to 45 degrees away from its starting orientation, the mean orientation estimation error increased to 11.0 degrees, as shown by the outer green region in Figure 5.13. From this, we conclude that the implemented capacitive sensor and pose estimator are most accurate when the relative orientation offset between the sensor and human limb is less than 30 degrees around the rotational axes. In Chapter 5.5.3, we show that this orientation estimation is accurate enough to enable a PR2 to sense and adapt to local changes in a person's limb orientation, such as around an elbow or knee joint, or due to human motion.

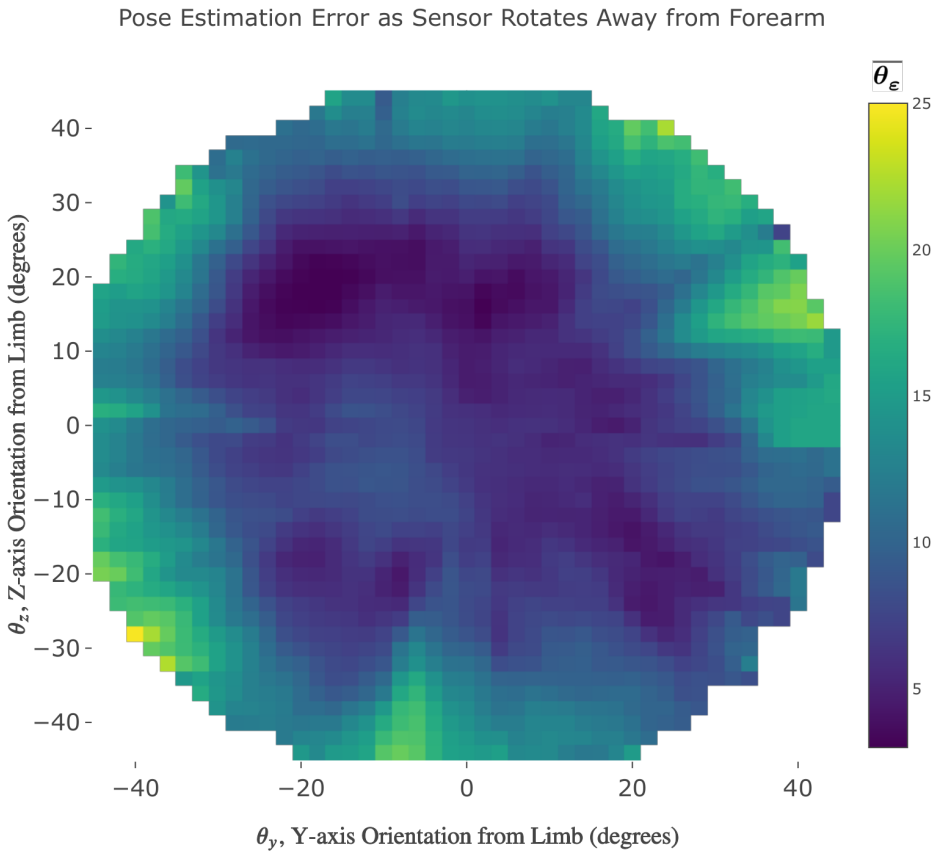


Figure 5.13: Mean estimation error $\overline{\boldsymbol{\theta}_{\boldsymbol{\epsilon}}}$ (in degrees) of the trained pose estimation model when the capacitive sensor was at various orientations facing away from a human forearm. The pose estimation error in each $2^\circ \times 2^\circ$ grid cell with center (y, z) is averaged over all data pairs $(\boldsymbol{c}_{t-49:t}, \boldsymbol{p}_t)$ near the cell where $\theta_y \in [y-3, y+3]$ and $\theta_z \in [z-3, z+3]$. This circular region has a radius of 45 degrees, wherein the robot would rotate the capacitive sensor around the Y- and Z-axes until it was oriented 45 degrees from the sensor's starting orientation above a participant's forearm, i.e. $||\boldsymbol{\theta}||_2 = 45$ degrees.

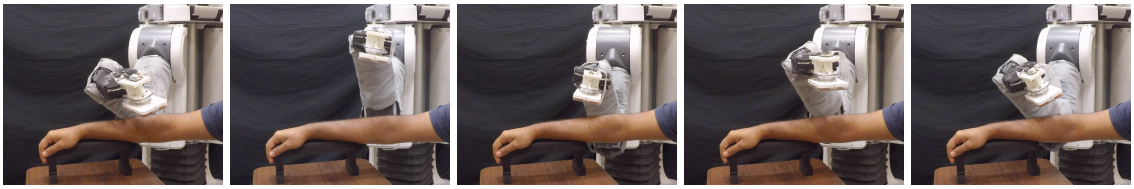


Figure 5.14: The training data collection process visualized for one of the 12 participants. The PR2 collected capacitance and pose data pairs $(\boldsymbol{c}_t, \boldsymbol{p}_t)$ while performing 50 randomized trajectories above the participant's forearm. We used these data to evaluate generalization performance of capacitive sensing across participants of different body size.

Figure 5.15: The training data collection process (detailed in the Appendix) as the PR2 moved the capacitive sensor over a participant's shin.

5.5.2 Generalization Across People

In this section, we evaluate the generalization performance of capacitive sensing for pose estimation. We do so by evaluating the accuracy of the proposed pose estimation model (see Chapter 5.4.4) for estimating the limb poses of all 12 participants. We then contrast this performance to models trained on capacitance measurements from multiple people with different body sizes.

For this experiment, we repeat the training data collection process, shown in Data Collection 1 (Appendix), with all participants. For each participant, the PR2 collected data pairs $(\boldsymbol{c}_t, \boldsymbol{p}_t)$ while performing $N = 50$ uniformly random linear end effector trajectories above each of the six locations along the arm and leg (wrist, forearm, upper arm, ankle, shin, and knee). In total, we collected over 590,000 capacitance measurements across all 12 participants, with approximately 99,000 capacitance measurements above each of the six limb locations. Figure 5.14 depicts several frames of this data collection process from one of the 12 participants as the PR2 moved the capacitive sensor above the participant's forearm. Figure 5.15 visualizes data collection above the participant's shin.

We temporally window the capacitance data into pairs $(\boldsymbol{c}_{t-49:t}, \boldsymbol{p}_t)$, which we feed

Table 5.1: Pose estimation error per limb location averaged over all 12 participants. D_y, D_z are in (cm). θ_y, θ_z are in (degrees).

	Average error per sensing axis			
	D_y (cm)	D_z (cm)	θ_y (deg)	θ_z (deg)
Wrist	2.5 ($\pm$2.6)	1.8 ($\pm$2.0)	8.6 ($\pm$6.9)	9.9 ($\pm$7.6)
Forearm	2.1 ($\pm$2.2)	1.6 ($\pm$1.6)	6.5 ($\pm$5.7)	9.2 ($\pm$7.1)
Upper arm	2.6 ($\pm$2.3)	1.9 ($\pm$1.6)	7.7 ($\pm$6.4)	10.3 ($\pm$7.7)
Ankle	2.0 ($\pm$1.8)	1.8 ($\pm$1.6)	8.1 ($\pm$6.6)	9.6 ($\pm$7.2)
Shin	2.2 ($\pm$2.1)	1.7 ($\pm$1.6)	6.8 ($\pm$5.9)	9.1 ($\pm$7.1)
Knee	2.2 ($\pm$2.1)	1.7 ($\pm$1.4)	8.2 ($\pm$6.5)	9.3 ($\pm$7.1)
Average	**2.27** cm	**1.75** cm	**7.65**°	**9.57**°

Table 5.2: Average error when training a pose estimation model on $n-1$ participants and evaluating on the heldout participant.

	D_y	D_z	θ_y	θ_z
Average	2.06 cm	1.39 cm	6.97°	8.85°

through our trained pose estimation model f. Table 5.1 depicts the pose estimation error of our model averaged over all 12 human participants for each of the sensing axes $(D_y, D_z, \theta_y, \theta_z)$. Although the pose estimation model was trained on capacitance data from only a single human participant, the model exhibits on average less than 2.5 cm of position estimation error and less than 10° of orientation error per axis. In practice, this model generalizes well enough to be used in a capacitive servoing control loop for performing assistive tasks around the human body (Chapter 5.5.4).

Similarly, we investigate the benefit of training capacitive pose estimation models on data from more than one human participant using leave-one-participant-out cross-validation. We retrain the pose estimation network (see Chapter 5.4.3) on capacitance measurements from 11 participants and evaluate pose estimation performance on data from the one held-out participant. We repeat this process by holding out each participant, resulting in 12 trained pose estimation models and corresponding heldout test sets. In Table 5.2, we report leave-one-participant-out results as the mean estimation error over all 12 models.

Table 5.3: Pose estimation error for the participants with the minimum and maximum measured limb circumference.

	Average error per sensing axis			
Limb (Circumference)	D_y	D_z	θ_y	θ_z
Wrist (13 cm)	2.1 cm	1.6 cm	7.2°	10.0°
Wrist (18 cm)	2.2 cm	1.4 cm	10.1°	12.4°
Forearm (22 cm)	1.7 cm	1.4 cm	6.6°	10.6°
Forearm (29 cm)	2.0 cm	1.4 cm	6.5°	10.4°
Upper arm (24 cm)	2.1 cm	1.6 cm	8.4°	11.7°
Upper arm (30 cm)	2.4 cm	1.9 cm	12.3°	11.5°
Ankle (20 cm)	2.2 cm	2.0 cm	8.3°	11.9°
Ankle (26 cm)	1.6 cm	1.6 cm	7.1°	10.0°
Shin (25 cm)	3.0 cm	2.4 cm	7.6°	9.1°
Shin (40 cm)	2.0 cm	1.7 cm	7.9°	8.7°
Knee (25 cm)	2.0 cm	1.5 cm	7.7°	9.2°
Knee (37 cm)	2.1 cm	1.4 cm	6.0°	8.6°

We observed similar pose estimation performance between a model trained on one participant (Table 5.1) and models trained on multiple participants (Table 5.2). With less than 0.5 cm difference in translational (D_y, D_z) error (difference between Table 5.1 and Table 5.2) and less than 1° difference in rotational (θ_y, θ_z) error, these results indicate that pose estimation models trained on capacitance measurements from one participant are likely to generalize to a larger population of people.

Another question that arises is how well estimation via capacitive sensing performs across both small and large sized limbs. Table 5.3 provides the pose estimation error of our model for individual participants with either the smallest or largest circumference for a given limb. Error values in Table 5.3 are averaged over all $N = 50$ data collection trials for a single participant and limb location. Limb circumferences ranged from 13 cm (wrist of the smallest participant) to 40 cm (shin of the largest participant).

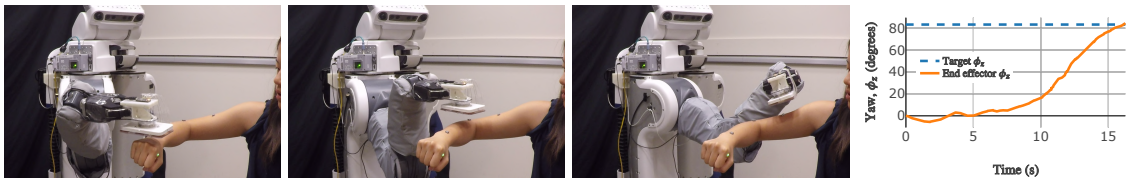


Figure 5.16: The PR2 used capacitive servoing to move its end effector proximally from the participant's hand to shoulder. The participant bent their elbow at an ∼90 degree angle. The right plot displays the yaw orientation ϕ_z of the capacitive sensor over time (determined by forward kinematics) and the target yaw orientation of the participant's upper arm (visually estimated using reflective markers on the arm). 3D positions of the infrared reflective markers on a participant's wrist, elbow, and upper arm *were not* provided to the robot during trials.

Despite not providing the pose estimation model with explicit information about limb circumference, results were relatively stable across the various limb sizes. Average error in lateral translation D_y ranged from 1.6 cm to 3 cm, whereas estimation error in vertical translation D_z ranged from 1.4 cm to 2.4 cm. Error in pitch orientation estimation θ_y ranged from 6.0° to 12.3° and yaw orientation error θ_z varied from 8.6° to 12.4°.

5.5.3 Capacitive Servoing Around Human Limbs

One strength of capacitive sensing is its ability to sense the contours along a human limb. For example, the process of sensing and moving a robot's end effector along a person's limb is the fundamental task required for several caregiving scenarios, including dressing and bathing assistance. In this section, we evaluate how a robot can use multidimensional capacitive servoing (Algorithm 4) to navigate around the contours of human limbs.

We design three tasks in which the robot must navigate around a static human limb at various poses: (1) *Bent Elbow*, participant arm is held parallel to the ground with a bent elbow (Figure 5.16), (2) *Forearm Tilt*, forearm and hand are tilted towards the ground while the elbow is bent (Figure 5.17), (3) *Bent Knee*, participant bends their leg while sitting on a chair (Figure 5.18).

In the first task, each participant held his or her arm in a static pose parallel to the ground with their elbow bent at either 0, 30, 60, 90, or 120 degrees. The robot positioned

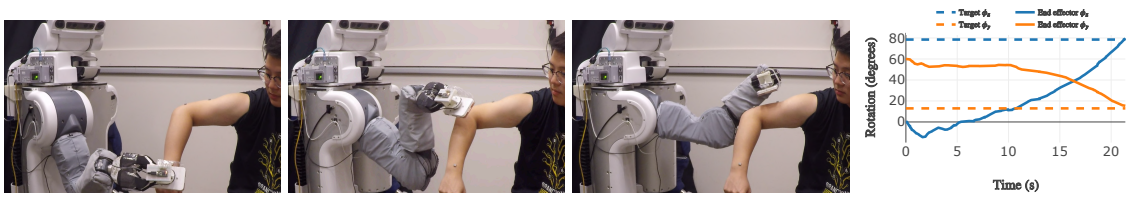


Figure 5.17: The PR2 used capacitive servoing to move proximally to a participant's shoulder around a tilted forearm and bent elbow. The participant bent their elbow at a right angle and tilted their forearm ∼60 degrees towards the ground. The plot displays the yaw orientation ϕ_z and pitch orientation ϕ_y (tilt) of both the robot's end effector and participant's upper arm (target).

Figure 5.18: The PR2 used capacitive servoing to control its end effector distally from a participant's thigh to ankle. The participant bent their knee ∼60 degrees by resting their foot on a stool. The plot shows the pitch orientation ϕ_y of the robot's end effector and the participant's shin (target) over time.

the capacitive sensor starting 5 cm above a participant's hand and used capacitive servoing to navigate the end effector and sensor around the bent elbow and up to a participant's shoulder. Figure 5.16 depicts a representative image sequence of the robot using capacitive servoing to follow the contours of a participant's arm while they hold a 90 degree elbow bend. The plot in Figure 5.16 shows the yaw orientation ϕ_z of the capacitive sensor over time (determined by the robot's forward kinematics) and the target yaw orientation of the participant's upper arm (determined by visually reflective markers on the arm). We observe that the robot was able to sense and rotate around the bent elbow to match the target orientation of the participant's upper arm by the end of the trial.

For the second task, each participant held their arm parallel to the ground with a 90 degree elbow bend, and then tilted their forearm and hand towards the ground at degree intervals of 0, 30, 60, and 90 degrees. Figure 5.17 shows an image sequence for capacitive servoing around a participant's arm as their forearm is tilted ∼60 degrees towards the ground. The corresponding plot in Figure 5.17 depicts the yaw ϕ_z and pitch ϕ_y orientation

Figure 5.19: An example in which capacitive servoing lost track of a participant's leg when attempting to rotate around a 90 degree bent knee.

Table 5.4: Task success averaged over all trials.

	Rotation of limb joint (degrees)				
Task	0	30	60	90	120
Bent Elbow	100%	100%	100%	100%	91.6%
Forearm Tilt	100%	100%	100%	100%	—
Bent Knee	100%	100%	100%	62.5%	—

of the capacitive sensor along with the target orientations determined by the pose of the participant's upper arm. We again observe that the robot's end effector was able to traverse the participant's arm and match the orientation of the upper arm by the end of the capacitive servoing trial.

During the final task, participants sat on a chair and rested their foot on a stool such that their knee was bent at 0, 30, 60, or 90 degrees. The robot began above a participant's thigh and then used capacitive servoing to navigate down towards their ankle. A sequence of this task is shown in Figure 5.18 where the participant had an approximately 60 degree bend at their knee. The associated plot shows the pitch ϕ_y of the sensor as it rotates around the knee to match the target orientation of the shin (visually estimated by reflective markers on the knee and ankle).

We conducted two human-robot trials for each combination of task and joint rotation, where a trial is defined successful if the capacitive sensor has navigated past the joint and is within 8 cm of the participant's limb towards the end of the trial. Table 5.4 provides the success rates for capacitive servoing across the three human limb tracking tasks. While capacitive servoing worked well for most tasks and limb poses, there are cases in which the

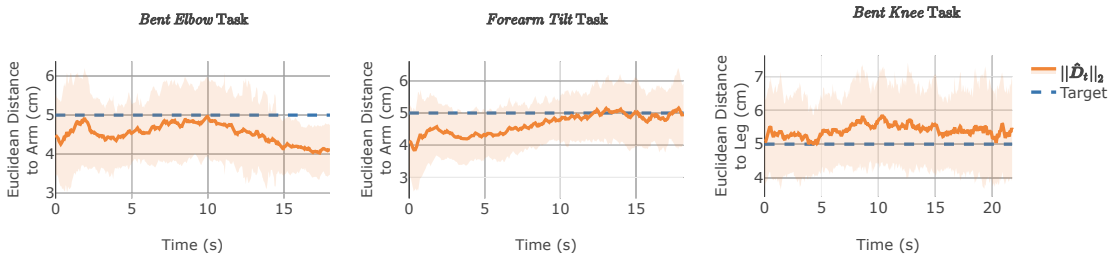


Figure 5.20: Estimated distance $||\hat{\boldsymbol{D}}_t||_2$ between the capacitive sensor and human limb, averaged over all participant trials and joint rotations for each task. During each trial, the robot is instructed to remain 5 cm above the participant's limb given $\boldsymbol{p}_{desired} = (0, 5, 0, 0)$ (see Chapter 5.4.5).

feedback controller would fail to track the limb contour correctly. An example of this is shown in Figure 5.19 where the robot struggled to rotate fast enough around a participant's 90 degree bent knee, resulting in the capacitive sensor navigating away from and losing track of the limb. When failures occurred for the *Bent Elbow* and *Bent Knee* tasks, they were mostly attributable to the feedback controller failing to rotate quickly enough around a sharp bend in a limb joint. In these cases, adaptive or more aggressive control algorithms may be able to rapidly adjust a robot's trajectories to more extreme human poses or human motion.

Figure 5.20 shows the estimated Euclidean distance $||\hat{\boldsymbol{D}}_t||_2$ between the capacitive sensor and human limb, averaged over all trials and joint rotations for each task. On average across all three tasks, the estimated capacitive sensor position remained within 1 cm from its target distance away from the human limb, namely $\boldsymbol{p}_{desired} = (0, 5, 0, 0)$.

5.5.4 Capacitive Servoing for Robotic Caregiving

In a prior study [77], we have demonstrated that capacitive servoing can enable a robot to provide assistance for two activities of daily living: dressing and bathing. In the first part of the study, a PR2 used capacitive servoing to adapt to arm movements made by four able-bodied participants while helping dress on the sleeve of a hospital gown. Participants performed vertical and lateral arm motions by translating their arm and bending their elbow such that their hand remained within 20 cm of their starting pose. Figure 5.21 depicts

Figure 5.21: The robot using capacitive servoing to sense and adapt to a participant's lateral arm motions during dressing assistance with a hospital gown sleeve. The PR2's end effector tracked both translational movements and orientation changes in the participant's arm pose in real time.

an image sequence of the PR2 using capacitive servoing to track a participant's lateral arm motions during dressing assistance. Despite arm motions, the estimated capacitive sensor pose $\hat{\boldsymbol{p}}_t$ remained on average within 3 cm from the desired pose $\boldsymbol{p}_{desired}$ above each participants' arm throughout the duration of each trial.

This prior research also investigated how capacitive servoing can enable a robot to assist with bathing, a task that requires continuous contact with a person's body. The robot wiped off a blue powder from the top surface of participants' arms and legs using a wet washcloth which we attached to the bottom of the capacitive sensor array. Because of the change in electrical conductivity between air and wet cloth, we repeated data collection and trained a new pose estimation model (Chapters 5.4.3 and 5.4.4) to better calibrate pose estimates for bathing assistance. Figure 5.22 demonstrates the robot using capacitive servoing to bathe a participant's arm. The capacitive sensor and washcloth moved proximally from the wrist to shoulder, while sensing and rotating around the elbow. In total, the robot succeeded to clean all powder off each participant's arm in 7 of the 8 trials from the study.

5.6 Discussion

Overall, capacitive servoing has shown to be a generalizable method for traversing a robot's end effector along the contours of human limbs and adapting to human limb motion. Yet, it is worth noting that the magnitude of capacitance measurements do vary depending on the circumference of a human limb. For both of the arm navigation tasks (*Bent Elbow* and *Forearm Tilt*) in Chapter 5.5.3, the capacitive sensor array was on average slightly less than

Figure 5.22: A bathing task during which the PR2 used a wet washcloth to clean off blue powder from a participant's arm. The participant's elbow was bent 90 degrees and their forearm was tilted 60 degrees towards the ground. Similar to the *Forearm Tilt* task from Chapter 5.5.3, the robot used capacitive servoing to traverse the participant's arm, while maintaining contact with the arm.

5 cm above the arm. However, the sensor was on average slightly greater than 5 cm from the leg during the *Bent Knee* task. This can be attributed to the larger surface area of a human leg. Based on the capacitance equation for a parallel plate capacitor $C = \frac{k\varepsilon_0 A}{d}$, the capacitance C is approximately proportional to the relative surface area A of the electrode and sensed object (limb) (where k, ε_0 are constants and d is the separation distance). Capacitance measurements typically have greater magnitude near the leg, causing the pose estimator to believe the sensor is closer to the limb than it actually is. For example, based on the data used to train our pose estimator (Chapter 5.4.4), capacitance measurements had a 44.5% increase in magnitude when the capacitive sensor array was 5 cm above the participant's leg as compared to above the arm. Future work could explore providing the pose estimator with a priori knowledge on the nearest limb segment (forearm, shin, thigh, etc.) or approximated limb circumference to help resolve this ambiguity.

Techniques for simulating capacitance signals around the human body may also serve as a way to improve capacitive servoing methods in future work. With the accessibility of physics simulation for physical human-robot interaction (see Assistive Gym [137]), simulated capacitance measurements can serve as a feature for safely learning intelligent robot controllers across a diverse array of human body shapes and sizes. As a step towards

simulated capacitive sensors, Clegg et al. [134] introduced a simulated multidimensional distance sensor for measuring the shortest Cartesian distance to the human body within a simulated robot-assisted dressing environment.

In addition to being mounted to assistive tools, as shown in this work, capacitive sensors also provide significant flexibility in where they are mounted on a robot. For robotic caregiving tasks where a robot's end effector is in close proximity to the human body, capacitive sensors can be adhered directly to the outer surface of a robot's gripper. For example, several prior studies have integrated capacitive sensors onto anthropomorphic robotic hands for object manipulation [138, 139].

In this chapter, we evaluate capacitive servoing for controlling a robot's end effector along human limbs. Limbs provide an assistive robot an inherent direction of motion for which to move the robot's end effector along (e.g. from hip to ankle). Further development of capacitive servoing may also be extended to navigating a robot around the human torso. Doing so would present new challenges in adapting to the large distribution of human torso sizes and geometries, and the fusion of additional sensory modalities to replace the directional information that is inherent to limbs.

5.7 Conclusion

This chapter presents a capacitive servoing control scheme for robots to interact with the human body, including estimating relative human limb pose, navigating along human limbs, and tracking human motion while providing physical assistance for robotic caregiving tasks.

We presented a design overview of multi-electrode capacitive sensors that can sense human limb pose when mounted to a robot end effector. Given a six-electrode capacitive sensor array, we also described a formal method for collecting capacitance data around human limbs to train human pose estimation models. Capacitive servoing couples these human pose estimates with feedback control, which enables a robot to sense and adapt to

human limb pose during physical human-robot interaction.

Through a human-robot study with 12 human participants, we investigated the sensing ranges and pose estimation accuracy of a multidimensional capacitive sensor and human limb pose estimator. The trained pose estimation model provided accurate 4D limb pose estimates when the capacitive sensor was within 15 cm away from the limb and up to 30° from the orientation of the limb. Our results also indicated that pose estimation models used in capacitive servoing generalize well across varying human body sizes. With leave-one-participant-out cross-validation, we observed that a pose estimation model trained on capacitance measurements from only one human participant performed on-par with models trained on data from multiple human participants. Capacitive servoing runs in real time, and in our studies was able to compute pose estimates of the human limb at frequencies over 100 Hz, using only the robot's on-board CPUs. As part of our evaluation, a PR2 robot used multidimensional capacitive servoing to move its end effector proximally and distally along a human limb, which is the fundamental task for many robotic caregiving scenarios, including dressing and bathing assistance.

CHAPTER 6
ASSISTIVE GYM: A PHYSICS SIMULATION FRAMEWORK FOR ASSISTIVE ROBOTICS

6.1 Research Summary

Autonomous robots have the potential to serve as versatile caregivers that improve quality of life for millions of people worldwide. Yet, conducting research in this area presents numerous challenges, including the risks of physical interaction between people and robots. Physics simulations have been used to optimize and train robots for physical assistance, but have typically focused on a single task. In this chapter, we present Assistive Gym, an open source physics simulation framework for assistive robots that models multiple tasks. It includes six simulated environments in which a robotic manipulator can attempt to assist a person with activities of daily living (ADLs): itch scratching, drinking, feeding, body manipulation, dressing, and bathing. Assistive Gym models a person's physical capabilities and preferences for assistance, which are used to provide a reward function. We present baseline policies trained using reinforcement learning for four different commercial robots in the six environments. We demonstrate that modeling human motion results in better assistance and we compare the performance of different robots. Overall, we show that Assistive Gym is a promising tool for assistive robotics research.

6.2 Introduction

In 2014, 27.2 percent, or 85 million, of people living in the United States had a disability [140]. About 17.6 percent, or 55 million people, had a severe disability. Autonomous robots that provide versatile physical assistance offer an opportunity to positively impact the lives of people who require support with everyday tasks, yet conducting this type of re-

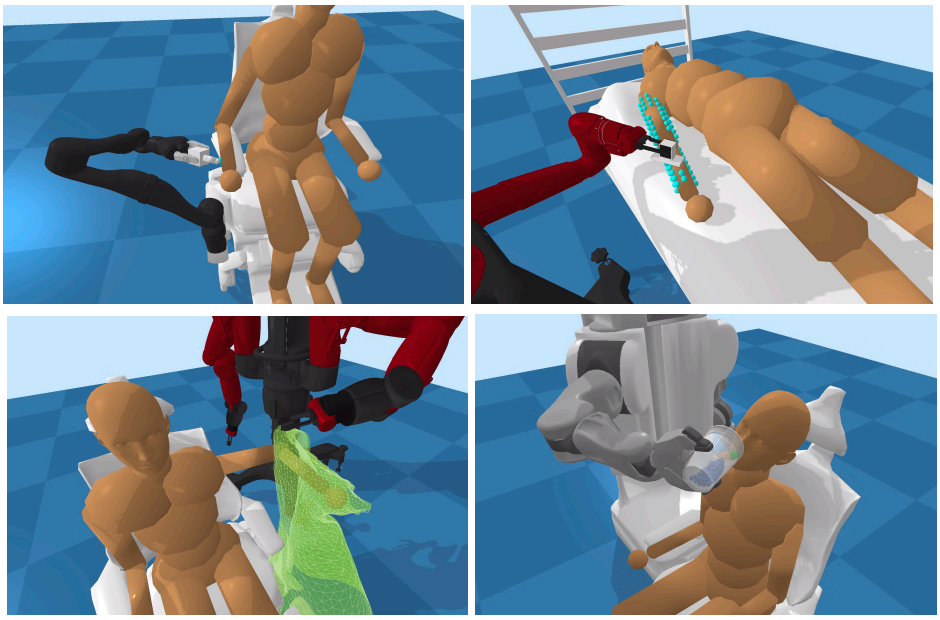


Figure 6.1: Four robots in Assistive Gym providing physical assistance. The four tasks include itch scratching, bed bathing, dressing, and drinking assistance.

search presents several challenges, including high costs and risks associated with physical human-robot interaction.

When compared to real-world robotic systems, physics simulation allows robots to safely make and learn from mistakes without putting real people at risk. Physics simulations can also parallelize data collection to perform thousands of human-robot trials in a few hours and provide models of people representing a wide spectrum of human body shapes, weights, and physical capabilities/impairments.

In this chapter, we present Assistive Gym[1], an open source physics-based simulation framework for physical human-robot interaction and robotic assistance. In comparison to existing robotic simulation environments, Assistive Gym places a strong emphasis on modeling the interaction between robots and humans (Figure 6.1), and builds off of prior research on how robots can provide intelligent physical assistance to people [141, 46, 77, 142, 143].

Assistive Gym is integrated into the OpenAI Gym framework. It provides an intuitive

[1] https://github.com/Healthcare-Robotics/assistive-gym

and familiar interface for developing control algorithms that enable robots to more intelligently interact with and assist people [144]. With this framework, we introduce physics-based environments where robots can assist people with six different tasks associated with activities of daily living (ADLs); including itch scratching, drinking, feeding, body manipulation, dressing, and bed bathing assistance—all of which are commonly requested tasks among adults who require physical assistance [64].

Assistive Gym presents several use cases for the research community. First, Assistive Gym can be used as a benchmark to compare control algorithms for robots that interact with people. Assistive Gym also provides the groundwork for researchers to develop environments and controllers for their own assistive tasks. Additionally, it allows researchers to design or compare robots for a given task.

Understanding a person's preferences for receiving care can be crucial for robots that wish to provide consistent and high quality care. We emphasize and model human preferences throughout the Assistive Gym framework, and we demonstrate how robots can learn to prioritize providing assistance that is consistent with a person's preferences.

We provide and evaluate baseline control policies for four commercial robots for each assistive task: the PR2, Sawyer, Baxter, and Jaco robots. Finally, there are often times when a person who requires assistance will have some limited motor functionality. In these scenarios, we show that by modeling human movement in simulation, robots learned to provide better assistance, with task success increasing by 30.4% on average. We model this as a co-optimization problem, in which both the human and robot are active agents that are simultaneously trained to achieve the same assistive task.

Through this work, we make the following contributions:

- We release a simulation framework, Assistive Gym, for physical human-robot interaction and robotic assistance.
- We introduce assistive robotic environments grounded in prior research for six activities of daily living.

- We present and analyze baseline robot controllers for each assistive task, and we show how Assistive Gym may be used to compare robots for assistive tasks.
- We show that by modeling human motion through co-optimization, robots can learn to provide improved assistance for a variety of tasks.
- We discuss modeling of human preferences in physical human-robot interaction simulations and we present a unified model of human preferences for assistive tasks.

6.3 Related Work

6.3.1 Simulation Environments

OpenAI Gym is a framework for learning control policies for simulated agents that includes a collection of benchmark problems, a common interface, and comparison tools. Its benchmark environments include Atari games and physics-based locomotion agents [144, 145, 146]. Assistive Gym builds on the OpenAI Gym framework and is intended to fulfill a similar role for the field of assistive robotics research.

Three physics engines commonly used for simulating robotics environments in OpenAI Gym are PyBullet [147], DART [148], and MuJoCo [149]. PyBullet, which we use to build our simulation environments, is a Python module for the open source Bullet Physics Engine that has been used for training and validating real robots using physics simulations [150, 151, 152, 153].

Several recent projects have begun presenting simulation environments for various robotic tasks including manipulation, navigation, or visual tasks [154, 155, 156]. Zamora et al. extended OpenAI Gym to ROS and Gazebo [157], which has been used by [158]. Similarly, Fan et al. introduced SURREAL, a physics simulation framework for robotic manipulation research that included six manipulation tasks, such as block stacking and bin picking [154]. In contrast to existing robotic simulation environments, Assistive Gym provides human-centric environments in which robots learn to directly help people in a variety

of tasks that have been shown to be important for quality of life.

6.3.2 Robotic Assistance

A number of works have explored robot-assisted feeding, using a variety of robots from wheelchair-mounted arms to mobile manipulators [159, 160, 161, 162, 163, 46]. Various robotic arms have also been used for robotic drinking assistance [142, 164]. Itch scratching assistance can be especially valuable for people who are unable to move their upper bodies due to disabilities [141, 165]. Researchers have also investigated robot assistance for bed bathing [75], including capacitive sensing to sense the human body [77].

Robot-assisted dressing has received significant focus in recent years [8, 10, 68, 29, 166, 7, 167, 11, 136]. Our prior research has also explored the use of physics simulations for learning control strategies for robot-assisted dressing [30, 143, 21, 70]. Clegg et al. presented a co-optimization approach for a robot and person to jointly learn a simulated dressing task [168]. We incorporate co-optimization into our evaluations, and we show how this approach can be extended to multiple different robots and assistive tasks. Recent work has also shown transfer between Assistive Gym and a real PR2 robot. Clegg et al. used Assistive Gym with reinforcement learning to learn a control policy for a real PR2 to dress a hospital gown onto a humanoid Meka robot [169].

Assistive Gym builds on many of the assistive robotics research discussed above, providing environments for robots to safely learn to interact with people with disabilities at scales much larger than currently available with robots in the real-world. Assistive Gym aims to facilitate assistive robotics research by providing a common simulation framework that can be used for training algorithms in many common assistive tasks. The framework can lower the barrier to development of algorithms, provide baselines for comparison, and provide canonical problems for the field.

6.4 Assistive Gym

Assistive Gym is a simulation framework with high level interfaces for building and customizing simulation environments for robots that physically interact with and assist people. Assistive Gym environments are built in the open source PyBullet physics engine [147]. PyBullet presents several benefits for simulating physical human-robot interaction, including real time simulation on both CPUs and GPUs, soft bodies and cloth simulation, and the ability to programmatically create robots and human models of varying shapes, sizes, weights, and joint limits. Assistive Gym integrates directly into the OpenAI Gym interface, allowing for the use of existing control policy learning algorithms, such as deep reinforcement learning.

6.4.1 Human Model and Robots

Assistive Gym provides support for four commercial collaborative robots that are commonly used for physical human-robot interaction. These include the PR2, Jaco, Baxter, and Sawyer robots, as shown in Figure 6.1.

We provide default male and female human models, with body sizes, weights, and joint limits matching published 50th percentile values [170]. The human model is programmatically generated, allowing for easy modification to the shape and properties of the human. Self collision between the various limbs and body parts of the human is also enabled. In total, the human model has 40 controllable joints, including an actuated head, torso, arms, waist, and legs.

Assistive Gym also provides models of several human limitations, including head and arm tremor, joint weakness, and limited range of motion. Given a target joint configuration, $\bar{\boldsymbol{q}}$, for the person's arm or head, we model tremor by adding an oscillating offset such that $\boldsymbol{q}_t = \bar{\boldsymbol{q}} + \epsilon(-1)^{t \bmod 2}$ where $\epsilon \in U(0, 20°)$ is sampled upon creation of the human model. We model joint weakness by multiplying the maximum torque, τ_{max}, of each human joint

by a scaling factor $\beta \in U(0.25, 1)$. We model limited range of motion by multiplying the pose-independent limits for each human joint, $\boldsymbol{l}_{min}$, which are negative, and $\boldsymbol{l}_{max}$, which are positive, by a scaling factor $\gamma \in U(0.5, 1)$. For the six assistive environments, the simulator randomly selects the male or female human model and applies one of these three limitations to the human.

6.4.2 Assistive Tasks (Environments)

As a part of Assistive Gym, we are releasing a suite of simulation environments for six tasks associated with activities of daily living (ADLs) [171], including:

- **Itch Scratching**: A robot holds a small scratching tool and must reach towards a random target scratching location along a person's right arm. The robot is rewarded for moving its end effector close to the target location, applying less than 10 N of force [70, 48] near the target, and performing small scratching motions near the target.

- **Bed Bathing**: A person lies on a bed in a random resting position while a robot must use a washcloth tool to clean off a person's right arm. The robot is rewarded for moving its end effector closer to the person's body and for wiping the bottom of the washcloth tool along the surface of the person's arm. We uniformly distribute markers (3 cm apart) around the person's right arm, for the robot to wipe off.

- **Drinking Water**: A robot holds a cup of small spheres (particles) representing water and must help a person drink this water. We randomize the starting orientation of the person's head. The robot is rewarded for moving the cup closer to the person's mouth, for tilting the cup, and for pouring water into the person's mouth.

- **Feeding**: A robot holds a spoon with small spheres representing food on the spoon and must bring this food to a person's mouth without spilling it. We randomize the starting orientation of the person's head. The robot is rewarded for moving the spoon closer to the person's mouth and placing food into the person's mouth.

- **Dressing**: A robot holds a hospital gown and must pull the sleeve of the gown up a person's left forearm and upper arm. The robot is rewarded for pulling the opening of the sleeve toward a person's hand, for pulling the sleeve onto a person's arm, and for pulling the sleeve along the central axis of the arm, towards the person's shoulder.
- **Arm Manipulation**: A person lies on a bed in a random resting position, with his/her right arm hanging off of the bed. A robot uses a scooping tool to lift a person's arm and place it on the bed next to his/her torso. The robot is rewarded for moving its end effectors towards the person's arm and for lifting the person's arm towards their torso.

For most of the assistive tasks, a robot holds a task relevant tool, such as a washcloth tool (see [77]) or a cup of water. Prior to the start of each task, the robot's base pose with respect to the person is optimized (section 6.4.4) and a random perturbation is added to the starting position of the robot's end effector (up to 5 cm along each global axis).

6.4.3 Realistic Human Joint Limits

With physical human-robot interaction in simulation, it is important to model realistic human joint limits as we want robots to learn to provide safe assistance that does not create discomfort. However, modeling joint limits can be difficult since they are pose-dependent—the range of motion for one joint is dependent on the configurations of other joints.

Akhter and Black presented a procedure of discrete operations to model realistic human joint limits [172]. Given a joint configuration for the human arm, $\boldsymbol{q}$, the authors provide a function, $C(\boldsymbol{q})$, which estimates whether the arm is in a valid pose. The authors fit their procedure on human motion capture data. Recently, Jiang et al. presented an approach to reduce the computational requirements of determining whether a human pose is valid using a fully connected neural network model trained on the human motion capture data [173]. We have incorporated this neural network model into Assistive Gym to model realistic human arm joint limits.

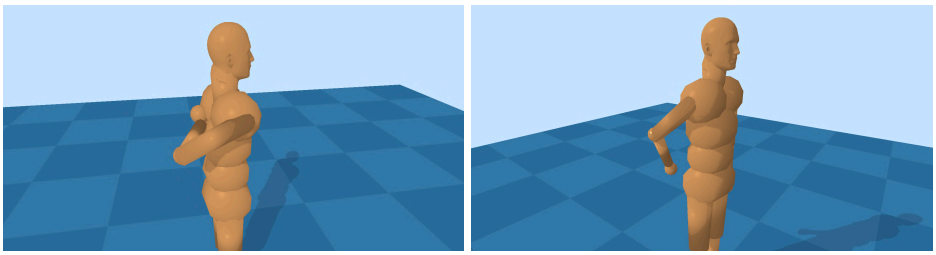


Figure 6.2: (Left) An attainable arm joint pose when joint limits are pose-independent. (Right) The realistic joint limits reached when the human attempts to move to the same pose with pose-dependent limits enabled.

Specifically, at each time step, t, during simulation, we compute $C(\boldsymbol{q}_{right,t})$ and $C(\boldsymbol{q}_{left,t})$ for both the right and left human arm, respectively. If the output from the model indicates that an arm is in an invalid pose, we then set the arm to the joint configuration from the previous time step, $\boldsymbol{q}_{,t-1}$, in which the arm was in a valid joint configuration. Additional details on the model and training procedure for realistic joint limits can be found in [173].

In Figure 6.2, we depict the impact of realistic human joint limits in Assistive Gym. Figure 6.2 (Left) shows an achievable arm pose that may be uncomfortable when all joints and joint limits are considered independent. Figure 6.2 (Right) shows an arm pose that is more likely to be comfortable, which is achieved when the realistic joint limit model is enabled and the human attempts to reach the previous configuration.

6.4.4 Robot Base Positioning

With a well-chosen base pose, a robot can better perform an assistive task despite model error, pose uncertainty, and other variations. We created a baseline method for selecting robot base poses with respect to a person (2D position and orientation) based on task-centric optimization of robot configurations (TOC), particularly joint-limit-weighted kinematic isotropy (JLWKI) and task-centric manipulability [174].

Assistive Gym randomly samples 100 base poses for the robot and attempts to find an inverse-kinematics (IK) solution to each goal end effector pose. It selects the robot base pose with a collision-free IK solution to the most goals. In case of ties, Assistive Gym

selects the base pose with highest JLWKI summed across all goal end effector poses.

JLWKI, as presented by Kapusta and Kemp [174], is a modification of manipulability from [175] and kinematic isotropy [176]. JLWKI computes a measure of dexterity using a weighting function on joint limits and applying it to kinematic isotropy. Additional details on JLWKI are present in [174].

The base pose for the PR2, Baxter, and Sawyer are optimized at the beginning of a simulation trial for all six assistive tasks in Assistive Gym. For the four tasks with a wheelchair, namely itch scratching, drinking, feeding, and dressing, we mount the Jaco to a fixed location on the wheelchair comparable to the wheelchair location recommended by Kinova who produces the Jaco arm [177]. For the bed bathing and arm manipulation tasks, where the person is lying on a bed, the Jaco robot is mounted to a nightstand next to the bed, and the nightstand pose is optimized.

6.4.5 Modeling Human Preferences

Understanding a person's preferences for receiving care may improve quality and consistency of care. For example, a person may prefer for the robot not to apply large forces to their body, or for the robot to perform slow and interpretable actions [178]. Prior research has noted the importance of considering human preference when planning robot actions such as navigation [179, 180], manipulation [181, 182, 178] around humans, and when providing physical assistance [183, 184]. Several researchers have investigated learning expert user preferences [185, 186, 187], and human preferences for robot trajectories [188, 189]. Other prior research on reinforcement learning has used human preference feedback to allow learning when reward functions are poorly defined [190] or to increase learning efficiency [191]. In comparison, we provide a set of baseline human preferences in Assistive Gym which are unified across multiple assistive tasks.

At each time step, Assistive Gym computes a human preference reward, $r_H(\boldsymbol{s})$, based on how well a robot is satisfying the person's preferences given the state of the system, $\boldsymbol{s}$.

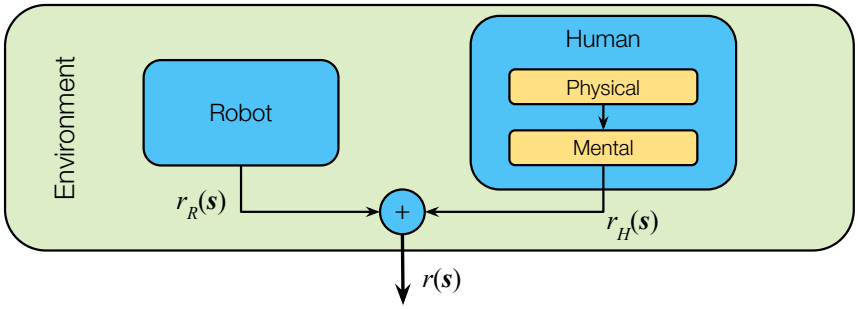


Figure 6.3: A model of a human-robot interaction environment that includes human preference. The physical state of a person (e.g. amount of force applied to the person) affects the person's mental model. Given an approximate mental state of the person, we can compute a human preference reward $r_H(\boldsymbol{s})$, which we combine with the robot's task success reward, $r_R(\boldsymbol{s})$.

As depicted in Figure 6.3, we combine this human preference reward with the robot's task success reward, $r_R(\boldsymbol{s})$, to obtain an overall reward, $r(\boldsymbol{s})$. By maximizing this reward a robot is able to learn control policies that are consistent with a person's preferences for receiving care. We use this same reward function, $r(\boldsymbol{s})$, when training both robot and human policies (Chapter 6.6.2). We define $r_H(\boldsymbol{s})$ as,

$$r_H(\boldsymbol{s}) \quad = \quad -\boldsymbol{\alpha} \cdot \boldsymbol{\omega} \odot [C_v(\boldsymbol{s}), C_f(\boldsymbol{s}), C_{hf}(\boldsymbol{s}), C_{fd}(\boldsymbol{s}), C_{fdv}(\boldsymbol{s}), C_d(\boldsymbol{s}), C_p(\boldsymbol{s})].$$

$\boldsymbol{\alpha}$ is a vector of activations in $\{0, 1\}$ depicting which human preferences are enabled for a given task, whereas $\boldsymbol{\omega}$ represents a vector of weights for each preference. For our environments, we use $\boldsymbol{\omega} = [0.25, 0.01, 0.05, 1.0, 1.0, 0.01, 0.01]$. $C_.(\boldsymbol{s})$ represents the cost of deviating from human preference in the state of the system, $\boldsymbol{s}$. We define the penalty terms as,

- $C_v(\boldsymbol{s})$, cost for high end effector velocities.
- $C_f(\boldsymbol{s})$, applying force away from the target assistance location (e.g. human mouth for drinking assistance).
- $C_{hf}(\boldsymbol{s})$, applying high forces near the target ($>$ 10 N).

- $C_{fd}(\boldsymbol{s})$, spilling food or water on the person.
- $C_{fdv}(\boldsymbol{s})$, food / water entering mouth at high velocities.
- $C_{d}(\boldsymbol{s})$, fabric garments applying force to the body.
- $C_{p}(\boldsymbol{s})$, applying high pressure with large tools.

For example, $C_v(\boldsymbol{s}) = \|\boldsymbol{v}_L\|_2 + \|\boldsymbol{v}_R\|_2$, where $\boldsymbol{v}_L$ and $\boldsymbol{v}_R$ are the velocities of the robot's left and right end effectors, respectively, or $\boldsymbol{v}_L = \boldsymbol{v}_R$ for single arm robots. We include a separate term for high pressure, as a robot must on occasion apply large forces distributed over a large surface area for tasks such as arm manipulation. Full equations for each term can be found in the source code.

6.5 Policy Learning and Control

Assistive Gym supports position control for commanding the robot. Actions for each robot's 7-DoF arm are represented as changes in joint positions, $\Delta P \in \mathbb{R}^7$ for single robot arms or $\Delta P \in \mathbb{R}^{14}$ for dual arm robots. Actions for a person include $\Delta P \in \mathbb{R}^{10}$ for the human arm, or $\Delta P \in \mathbb{R}^4$ for the head. We limit the strength of each robot actuator to reduce the likelihood of a robot learning policies that apply high forces to the person. This limitation could potentially be removed in future iterations using techniques, such as curriculum learning, to balance between making task success and satisfying human preferences [168]. In addition, prior research has shown that real robots can provide assistance based on what they have learned in simulation [143, 70], yet further work will be needed to enable real robots to benefit from controllers learned in Assistive Gym.

At each time step, the robot records observations from the state of the system, executes an action according to a control policy, and then receives a reward. We give observations to the robot in accordance to observations that can be obtained in a real-world assistive robotics scenario. This includes, the 3D position and orientation of the robot's end effector,

the 7D joint angles of the robot's arm, forces applied at the robot's end effector, and 3D positions of task relevant human joints, such as the wrist, elbow, shoulder, or the position and orientation of a person's head. Joint positions of a real human body can be obtained using existing approaches, such as with a single image using OpenPose, or the pose of a person on a bed using a pressure sensing mat [192, 67]. All positions are defined with respect to the robot's torso, or base position.

In this work, we use a deep reinforcement learning technique, proximal policy optimization (PPO), to learn control policies for robotic assistance. PPO is a policy gradient algorithm used across a number of contexts, from Atari games to real-world quadruped robot locomotion [193, 150]. We follow the original policy representation as presented in [193], using a fully-connected neural network with two hidden layers of 64 nodes, and tanh activations. For co-optimization in Chapter 6.6.2, we train two policies concurrently, for the robot and active human, with both optimized using a shared reward, $r(\boldsymbol{s})$ (Chapter 6.4.5). Recent work has transferred a PPO policy trained entirely in Assistive Gym to a real PR2 robot for an assistive dressing task [169]. Future work can further validate how controllers derived in Assistive Gym can transfer to real assistive robots while ensuring safety [194, 195].

6.6 Evaluation

In the following sections, we present and analyze baseline control policies for the four robots assisting with the six assistive tasks. For each task, we train robot controllers using PPO, as described in Chapter 6.5. We train each policy using 36 concurrent simulation actors for a total of 10,000,000 time steps, or 50,000 simulation rollouts (trials). Each simulation rollout consists of 200 time steps (20 seconds of simulation time at 10 time steps per second), where a policy can execute a new action at each time step. We perform a 10 epoch update of the policy after each actor completes a single rollout (i.e. every 7,200 time steps). We trained all policies using Amazon Web Services (AWS) with a 36

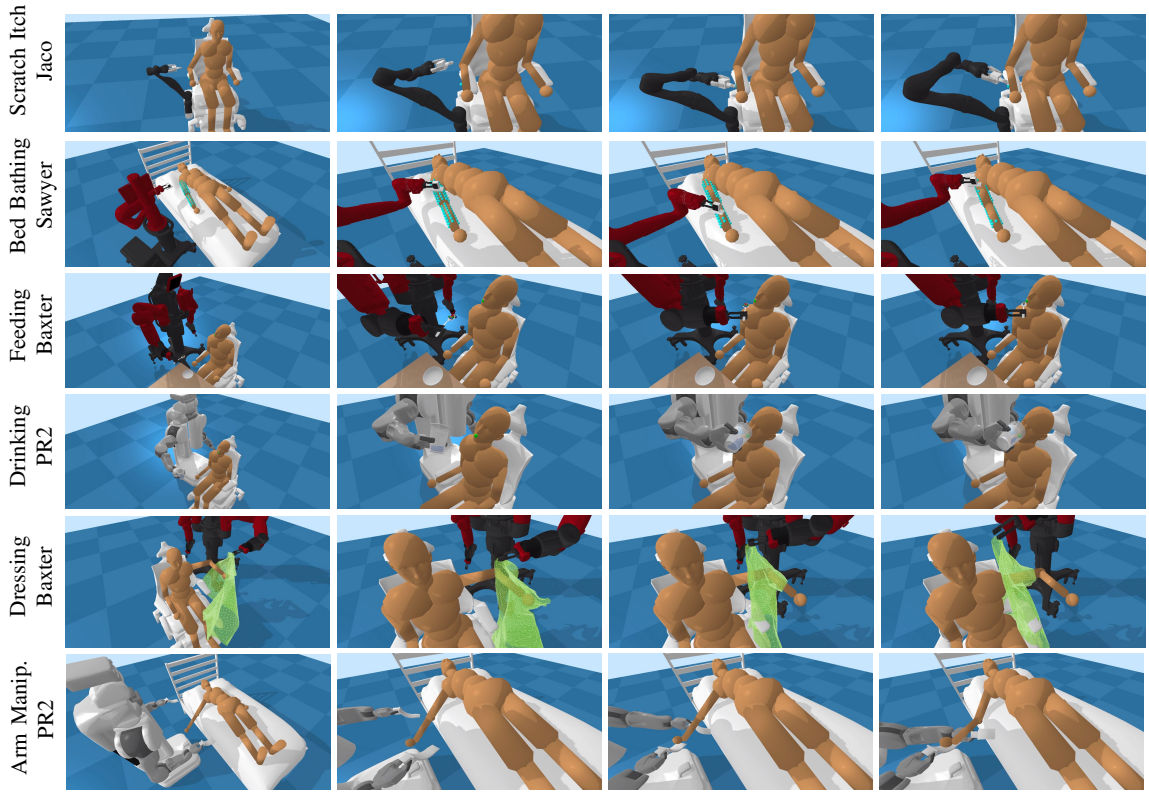


Figure 6.4: Image sequences from executing trained robot policies for each of the six assistive tasks when the person holds a static pose.

vCPU machine. Training times varied from 2 hours (itch scratching) to 19 hours (drinking). Training a policy for dressing assistance took ∼6 days due to simulating dynamic cloth.

6.6.1 Robotic Assistance

Similar to prior assistive robotic studies, we consider the scenario where a robot provides active physical assistance to a person who attempts to hold a static body pose [70, 196].

For each robot, we used PPO to train a separate control policy for each of the six assistive tasks. This amounts to a total of 6×4 = 24 unique policies. We then repeat each policy training with 3 random seeds and select the best of the three policies, based on the average reward achieved over the last 10 policy updates (the last 360 simulation rollouts).

Figure 6.4 shows image sequences for select robots executing their learned policies over all six of the assistive tasks. We note that for the first four tasks (itch scratching, bed

Table 6.1: Average reward for 100 trials with a **static human**. Task success on 100 trials for the robot with the highest reward.

Task	PR2	Jaco	Baxter	Sawyer	*Success*
Itch Scratching	55.1	**280.8**	225.4	136.8	**54%**
Bed Bathing	86.7	104.4	88.4	**109.0**	**24%**
Feeding	100.5	83.8	**108.5**	95.6	**88%**
Drinking	182.5	85.7	263.3	**436.0**	**72%**
Dressing	**11.5**	-17.0	5.6	-27.6	**26%**
Arm Manipulation	**-162.4**	-177.5	-228.1	-210.6	**8%**

bathing, feeding, and drinking), PPO was able to learn reasonable control policies for all four robots, with varying levels of performance between the robots. However, both the dressing and arm manipulation tasks remain challenging for all four of the robots. For dressing assistance, the robots struggled to pull the hospital gown up a person's forearm and upper arm, yet could on occasion when the opening of the gown sleeve was randomly placed close to the person's fist, as seen in Figure 6.4. For arm manipulation assistance, policy optimization had difficulties learning to lift a person's arm back onto the bed while also adhering to the person's preferences. The trained policies often had the robot use the thin edge of the tool to lift the person's arm, resulting in high pressure applied to the person and subsequently a large negative reward.

Assistive Gym also provides an opportunity to compare robot platforms in terms of their ability to provide physical assistance to people. To compare robots, we held all parameters and settings for PPO and the simulation environments constant. Given a trained control policy for a robot and assistive task, we evaluated the policy over 100 simulation rollouts of the task. Table 6.1 lists the average reward each robot achieved over the 100 rollouts for each task. Assistive Gym also defines task success for each task, and we include success rates for the best performing robot in Table 6.1.

Overall, we observed that there are significant opportunities for improvement for all of the robots when providing assistance to a static human across the six tasks. In addition, these results can help compare the physical capabilities of each robot. For example, the

PR2's shorter arm span made some tasks more difficult, such as itch scratching, which often requires reaching around the person's arm.

6.6.2 Collaborative Assistance

In the prior sections, we demonstrated how robots can learn to assist a person while the person holds a static pose. However, there are often scenarios in which a person who requires assistance will have some limited motor functionality. For example, a person may have a full range of motion of their arms, but suffer from large tremors, or conversely, may have fine motor control, but suffer from muscle weakness or a limited range of motion. In these cases, a person may prefer to actively help the robot accomplish its task, rather than hold a static pose. This is in part due to the assumption that people will be collaborative while receiving assistance, as they directly benefit from the assistance a robot provides.

We model human motion using co-optimization, in which both the human and robot are active agents that are trained simultaneously. We train separate control policies for the human and robot using PPO. Both the robot and human share the same reward function, but have different observation and action sets. For example, the observations for the robot include the robot's joint angles, whereas the observations for the human include the human's joint angles (proprioception).

This is inspired by work from Clegg et al., who explored co-optimization between a simulated KUKA IIWA robot and active human for robot-assisted dressing [168, 169]. Their work found that co-optimization can lead to more realistic human motions and improved assistance from the robot. Building upon their initial findings, we apply co-optimization across a wider variety of robots and assistive tasks. However, we note that co-optimization does not guarantee that the simulated human controller learns realistic human motions. Future work can investigate improved models of realistic human motion during assistive scenarios. Similar to the previous evaluations, we hold all parameters constant between the four different robots. We give the person's arm and head just enough motor strength to lift

Table 6.2: Average reward for 100 trials with an **active human**. Task success on 100 trials for the robot with the highest reward.

Task	PR2	Jaco	Baxter	Sawyer	*Success*
Itch Scratching	80.9	**443.2**	83.3	131.2	**68%**
Bed Bathing	90.2	**193.6**	175.5	166.2	**81%**
Feeding	**122.8**	106.1	108.3	112.5	**99%**
Drinking	**493.4**	402.6	466.8	464.0	**79%**
Dressing	-1.3	13.0	30.0	**56.9**	**89%**

Figure 6.5: Image sequence of a human policy rotating the person's arm so that the Jaco robot can better scratch an itch underneath their arm.

up against gravity. With co-optimization, we retrain policies for all of the robots and tasks, again choosing the best trained policy from three random seeds. Each robot policy has an associated human policy learned during co-optimization. We then evaluated each trained human and robot policy over 100 rollouts for a task. Table 6.2 depicts the average reward each robot achieved over the 100 rollouts when assisting an active human. Note that we do not evaluate the arm manipulation task, since an active human is often able to lift their arm back onto the bed, unaided by the robot.

Overall, we observed that in almost all cases, a robot is able to provide better assistance and achieve a higher reward when learning to assist an active human who performs collaborative motions. On average, the best performing robot achieved a 30.4% increase in task success when compared to assisting a static human. This is apparent when the robot assists with difficult to reach tasks, such as scratching underneath the upper arm, where an active human can rotate their arm to make it easier for the robot to reach and scratch the target location, as shown in Figure 6.5. Additional examples and results can be can be found in the supplementary video.

6.7 Conclusion

We presented Assistive Gym, an open source physics simulation framework for assistive robotics. Assistive Gym focuses on physical interaction between robots and humans. It models human physical capabilities and preferences for receiving assistance. We have provided baseline policies for four robots and six assistive tasks. We also demonstrated the use of Assistive Gym for benchmarking, for developing environments for assistive tasks, and for comparing robots. Overall, we have shown that Assistive Gym is a promising open source framework for the development of autonomous robots that can provide versatile physical assistance.

CHAPTER 7

ASSISTIVE VR GYM: INTERACTIONS WITH REAL PEOPLE TO IMPROVE VIRTUAL ASSISTIVE ROBOTS

7.1 Research Summary

Versatile robotic caregivers could benefit millions of people worldwide, including older adults and people with disabilities. Recent work has explored how robotic caregivers can learn to interact with people through physics simulations, yet transferring what has been learned to real robots remains challenging. Virtual reality (VR) has the potential to help bridge the gap between simulations and the real world. We present Assistive VR Gym (AVR Gym), which enables real people to interact with virtual assistive robots. We also provide evidence that AVR Gym can help researchers improve the performance of simulation-trained assistive robots with real people. Prior to AVR Gym, we trained robot control policies (*Original Policies*) solely in simulation for four robotic caregiving tasks (robot-assisted feeding, drinking, itch scratching, and bed bathing) with two simulated robots (PR2 from Willow Garage and Jaco from Kinova). With AVR Gym, we developed *Revised Policies* based on insights gained from testing the Original policies with real people. Through a formal study with eight participants in AVR Gym, we found that the Original policies performed poorly, the Revised policies performed significantly better, and that improvements to the biomechanical models used to train the Revised policies resulted in simulated people that better match real participants. Notably, participants significantly disagreed that the Original policies were successful at assistance, but significantly agreed that the Revised policies were successful at assistance. Overall, our results suggest that VR can be used to improve the performance of simulation-trained control policies with real people without putting people at risk, thereby serving as a valuable stepping stone to real robotic

assistance.

7.2 Introduction

Robotic assistance with activities of daily living (ADLs) could increase the independence of people with disabilities, improve quality of life, and help address pressing societal needs, such as aging populations, high healthcare costs, and shortages of healthcare workers [197, 198].

Physics simulations provide an opportunity for robots to safely learn how to physically assist people. Yet, the reality gap between physics simulations and real assistance poses several challenges. By bringing real people into the robot's virtual world, virtual reality (VR) has the potential to serve as a stepping stone between simulated worlds and the real world. For assistive robots, the person receiving assistance is an extremely important and complex part of the environment. As we show through a formal study, enabling real people to interact with a virtual robot can quickly reveal deficiencies through both objective and subjective measures of performance.

To facilitate the use of VR in the development of assistive robots, we present Assistive VR Gym[1] (AVR Gym) an open source framework that enables real people to interact with virtual assistive robots within a physics simulation (see Figure 7.1). AVR Gym builds on Assistive Gym, which is an open source physics simulation framework for assistive robots that models multiple assistive tasks [199]. AVR Gym enables people to interact with virtual robots without putting themselves at risk, which is especially valuable when evaluating controllers that have been trained in simulation with virtual humans.

As confirmed through a formal study with eight participants, our use of AVR Gym enabled us to identify significant, unexpected shortcomings in the simulation-trained baseline control policies that were originally released with Assistive Gym. Moreover, we were able to use AVR Gym to improve the control policies by discovering that the simulated humans

[1] https://github.com/Healthcare-Robotics/assistive-vr-gym

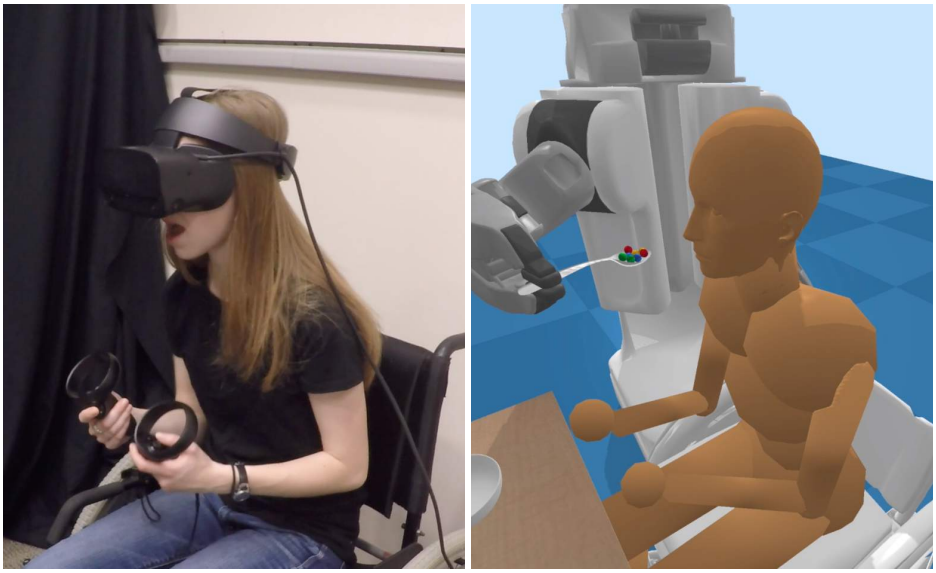


Figure 7.1: A participant using virtual reality to interact with a PR2 robot that has learned how to provide feeding assistance through physics simulation.

did not adequately represent the biomechanics of real people. Improving the simulated humans used to train control policies, resulted in Revised control policies that dramatically outperformed our Original policies.

In this chapter, we make the following contributions:

- Present Assistive VR Gym (AVR Gym), an open source framework that enables real people to interact with virtual assistive robots.
- Present experimental methods, including objective and subjective measures, to evaluate virtual assistive robots.
- Provide evidence that VR can be used to improve performance of simulation-trained robot control policies.
- Provide evidence that biomechanical models significantly impact policy performance and can be assessed and improved via VR.

7.3 Related Work

7.3.1 Physically Assistive Robotics

A significant amount of research has been conducted on real robotic systems for providing physical assistance to people. For example, researchers have demonstrated how table-mounted robotic arms can provide drinking assistance to people using a force sensing smart cup [164] and a brain-machine interface for shared-autonomy [200]. Similarly, robots have shown promising results for providing feeding assistance [201, 160, 202, 163, 161, 203]. Prior research has also investigated robotic assistance for bed bathing [75, 77], which can be especially valuable for people who are confined to a bed due to disabilities or injuries. Robots also present an opportunity to help individuals in getting dressed with a variety of garments. Robot-assisted dressing has been demonstrated on several robotic platforms for assisting both able-bodied participants and real people with disabilities [8, 204, 166, 205, 66, 77, 68, 29, 167, 11].

Despite these promising results in physical robotic assistance, it remains challenging to design robotic systems that can assist with multiple tasks across a wide spectrum of human shapes, sizes, weights, and disabilities. This is due in part to difficulties and costs associated with evaluating assistive robotic systems across a large distribution of people. Physics simulation presents an opportunity for robots to learn to safely interact with people over many tasks and environments. For instance, researchers have demonstrated the use of physics simulation for learning robot controllers for robot-assisted dressing tasks [30, 21, 70, 133]. Clegg et al. has also used Assistive Gym to learn control policies for a real PR2 robot to dress an arm of a humanoid robot with a hospital gown [134].

7.3.2 Virtual Reality

We use virtual reality to evaluate and improve simulation-trained assistive robots with real people. Studies have provided evidence that virtual reality can offer people both a sense of

presence and embodiment [206, 207]. Accordingly, VR provides an opportunity to assess important attributes for human-robot interaction (HRI), including proxemics, legibility of motion, and embodiment [208, 209].

Virtual reality has been widely used as a tool to improve robot performance across an assortment of tasks. A common use for virtual reality is for teleoperation of robots for dexterous manipulation tasks [210, 211]. This virtual reality teleoperation has also been used to collect high-quality robotic manipulation demonstrations [212, 213, 214]. Within robotic rehabilitation, virtual reality has been used to enhance rehabilitation training and assess rehabilitation metrics in real time [215, 216, 217, 218, 219].

Across these works, virtual reality has often been used for people to take control of robots; to provide demonstrations and to train robots for performing various tasks. In this chapter, we show that virtual reality can be used to safely evaluate and improve simulation-trained assistive robot controllers with real people.

7.4 Assistive VR Gym

AVR Gym builds on Assistive Gym and is implemented using the open source PyBullet physics engine [147]. We connect assistive environments from Assistive Gym into virtual reality with vrBullet, a virtual reality physics server that uses the OpenVR API. We use the Oculus Rift S virtual reality head-mounted display, which uses inside-out tracking, and two Oculus Touch controllers, each of which provide 3-DoF position and 3-DoF orientation tracking. Figure 7.2 depicts the virtual reality setup with 3rd and 1st person perspectives of the virtual environment.

The simulated human model in virtual reality has 20 controllable joints including two 7-DoF arms, a 3-DoF head, and a 3-DoF waist. Using the Oculus' inside-out tracking, we can estimate the height of each human participant as the difference from the headset to the ground. Given this height estimate, we modify the height of the simulated human body to match that of the real human participant. We then used the VR headset to align the

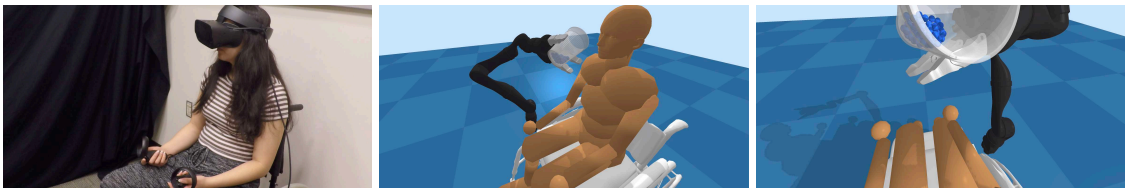


Figure 7.2: (Left) Human participant using the virtual reality interface to receive drinking assistance from a simulated robot. (Middle) 3rd person perspective of the simulated human model that the human participant is controlling. (Right) 1st person perspective of what a participant observed in virtual reality.

head and waist joints of a virtual human model to the person's pose. At each time step, we queried the headset for its 3D Cartesian position, $\boldsymbol{\chi}$, and 3D orientation, $\boldsymbol{\theta} = (\theta_r, \theta_p, \theta_y)$. We directly set the roll, pitch, and yaw joints of the simulated human head to align with the orientation, $\boldsymbol{\theta}$, of the headset.

To align the waist pose, we computed the appropriate angles needed such that the simulated human had the same 3D Cartesian head position as the real person. We first observed the fixed 3D position for the center of a person's waist, $\boldsymbol{W}$, given as a fixed offset above the wheelchair or hospital bed. Let $\boldsymbol{\psi} = (\psi_x, \psi_y, \psi_z) = \boldsymbol{\chi} - \boldsymbol{W}$ be the 3D vector from the center of a person's waist to the center of their head. We then computed the roll, pitch, and yaw $(r_{\boldsymbol{\psi}}, p_{\boldsymbol{\psi}}, y_{\boldsymbol{\psi}})$ orientations for the waist of the simulated human according to:

$$
\begin{aligned}
r_{\boldsymbol{\psi}} &= \text{atan2}(\psi_y, \psi_z) \\
p_{\boldsymbol{\psi}} &= \text{atan2}(\psi_x \cdot \cos(r_{\boldsymbol{\psi}}), \psi_z) \\
y_{\boldsymbol{\psi}} &= \text{atan2}(\cos(r_{\boldsymbol{\psi}}), \sin(r_{\boldsymbol{\psi}}) \cdot \sin(p_{\boldsymbol{\psi}}))
\end{aligned}
$$

Finally, we observe that people often prefer to use some waist motion when looking left or right. To account for this, we distributed some of the measured head yaw orientation to the waist. Specifically, we set the yaw orientation of the head to be $0.7(\theta_y - y_{\boldsymbol{\psi}})$, and the yaw orientation of the waist joint to be $0.3(\theta_y - y_{\boldsymbol{\psi}})$.

As shown in Figure 7.2, participants also hold two Oculus Touch controllers, which we use to control both arms of the simulated human in virtual reality. At each time step,

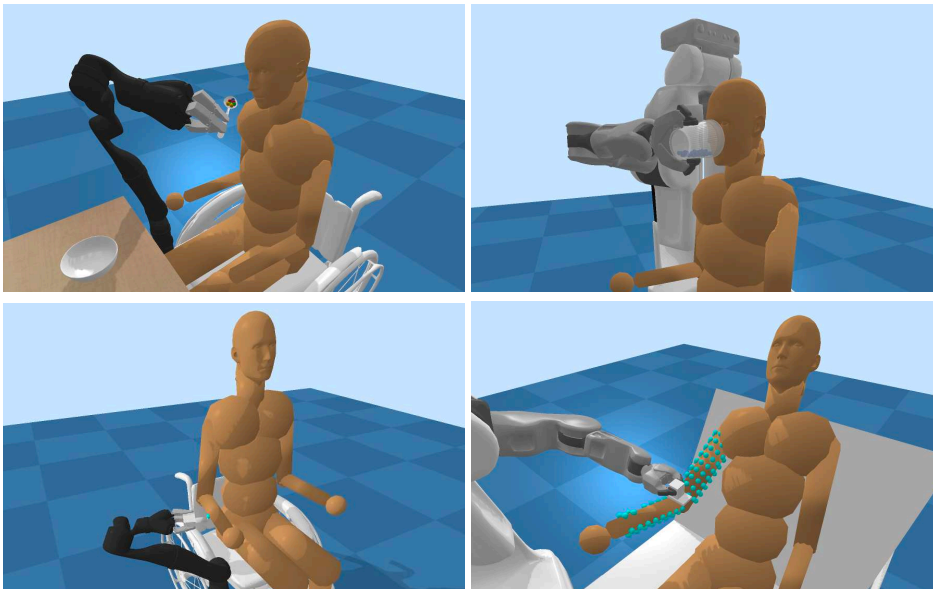


Figure 7.3: Assistive environments in physics simulation with the Jaco and PR2 robots. (Top row) Feeding and drinking assistance. (Bottom row) Itch scratching and bed bathing assistance.

we queried each handheld controller for its 3D Cartesian position and orientation in global space. Using inverse kinematics with a damped least squares method, we compute the 7-DoF arm joint angles necessary for the hand of the simulated person to have the same position and orientation as measured by the controller. Since we are optimizing for a 7-DoF arm pose using a 6-DoF controller pose measurement, this inverse problem is ill-posed, which implies that stable solutions are not guaranteed. However, we have found this inverse kinematics optimization to work well in practice, with only minor offsets in the estimated pose of a person's full arm.

7.4.1 Test Environments

AVR Gym builds upon four physics-based assistive environments from Assistive Gym. A key aspect of achieving realism for users was to match the simulated wheelchair and bed to a real wheelchair and bed. Each of these environments, shown in Figure 7.3, is associated with activities of daily living (ADLs) [171], including:

- **Feeding**: A robot holds a spoon and aims to move the spoon to a person's mouth. We

place small spheres on the spoon, representing food. The person sits in a wheelchair during assistance.

- **Drinking Water**: A robot holds a cup containing small particles that represent water. The robot aims to move the cup toward the person's mouth and tilt the cup to help the person drink the water. The person sits in a wheelchair during assistance.

- **Itch Scratching**: A robot aims to scratch a randomly generated location along the person's right arm. The robot grasps a scratching tool in its left end effector. The person sits in a wheelchair during assistance.

- **Bed Bathing**: The robot holds a simulated washcloth tool while the person lies on a hospital bed in a randomly generated resting pose. The robot aims to move the washcloth around the person's right arm in order to clean the arm.

These four assistive environments in AVR Gym are nearly identical to the original environments presented in [199], including the same male and female human models, realistic human joint limit models, robot base pose optimization, reward functions with human preferences, and task goals.

7.5 Simulation-Trained Robot Control Policies

Prior to developing AVR Gym, we trained eight robot control policies solely in simulation for the four caregiving tasks (robot-assisted feeding, drinking, itch scratching, and bed bathing) and two collaborative robots (the PR2 from Willow Garage and the 7-DoF Jaco (Gen2) arm from Kinova). These policies were the baseline policies released with Assistive Gym and described in the corresponding chapter [199]. We trained slightly modified versions of these eight robot control policies to run in AVR Gym, which we refer to as the *Original Policies*.

7.5.1 Original Policy Training

For each assistive task and robot, we follow the same training procedure presented in [199]. At each time step during simulation, the robot executes an action and then receives a reward and an observation based on the state of the world. Actions for a robot's 7-DoF arm are represented as changes in joint angles, $\Delta P \in \mathbb{R}^7$. The PR2 uses only its right or left arm depending on the assistive task. The observations for a robot include the 7D joint angles of the robot's arm, the 3D position and orientation of the end effector, and the forces applied at the end effector. The robot's observation also includes details of task-relevant human joints, including 3D positions of the shoulder, wrist, and elbow during the bed bathing and itch scratching assistive tasks, and the position and orientation of the person's head during feeding and drinking assistance. Both the Jaco and PR2 robots use the same observation and reward functions during training.

Our Original policies were trained using proximal policy optimization (PPO) [193], which is an actor-critic deep reinforcement learning approach that has recently shown success in assistive robotic contexts [168, 134, 199]. These policies are modeled using a fully-connected neural network with two hidden layers of 64 nodes and tanh activations. We train policies for 50,000 simulation rollouts (trials), where each rollout consists of 200 consecutive time steps (20 seconds of simulation time at 10 time steps per second). Prior to each simulation rollout, we randomly initialize the simulated person's arm pose for itch scratching and bed bathing tasks, and we randomize the head orientation for feeding and drinking tasks. Once initialized, the human holds a static pose throughout the entire rollout. Each policy is trained with default male and female human models, with heights, body sizes, weights, and joint limits matching published 50th percentile values [170]. Note that these control policies are not temporal models and do not consider observations from previous time steps in a rollout. As such, these models are not affected by the specific nature of human motion, allowing us to evaluate these policies with real people without needing to model realistic human motion in simulation.

7.5.2 Revised Policies

During early pilot studies in AVR Gym with lab members, we observed that the Original policies exhibited unexpected deficiencies and poor performance when providing assistance to human participants in virtual reality. Most notable were failures in the itch scratching and bed bathing assistance tasks, where the robots would fail to move their end effectors closer to the person's body. Through iterative investigation and development with real people in AVR Gym, we developed the *Revised Policies*, based on the key discovery that the biomechanical models of simulated people we used to train the Original policies significantly differed from the biomechanics of real people. The two leading factors were variation among human heights and waist bending movements, wherein the Original policies were trained on male and female models with fixed heights and were not trained on any variation among the human waist joints.

Given these findings, we modified each simulation environment such that the simulated human biomechanics better match people in VR. Specifically, we randomized the simulated human torso height and initial waist orientation before each training trial began. The person's torso height, measured from hipbone to center of the head, was uniformly sampled from 50 cm to 70 cm for male models and 44 cm to 64 cm for female models. We then randomized the simulated person's initial three kinematic waist joints to angles within the range of (-10°, 10°). With these improved biomechanical simulations, we trained a new set of eight *Revised Policies*, one for each robot and assistive task, using the same training process discussed in Chapter 7.5.1.

When evaluated with simulation humans, these Revised policies achieve similar rewards and task success rates as the Original policies. However, as confirmed by our formal study (Chapter 7.6), the Revised policies overcame the limitations of the Original policies and performed significantly better across both subjective and objective metrics when evaluated with real people in VR.

7.6 Formal Study

We conducted a formal study with eight participants to evaluate the performance of the Original policies and the Revised policies in terms of objective and subjective measures. In addition, we performed a posthoc analysis of biomechanical differences between the simulated humans used to train the Original policies, the simulated humans used to train the Revised policies, and the real humans who participated in our study.

7.6.1 Experimental Procedure

We conducted an experiment with eight able-bodied human participants (four females and four males). We obtained informed consent from all participants and approval from the Georgia Institute of Technology Institutional Review Board (IRB). We recruited participants to meet the following inclusion/exclusion criteria: 18 years of age or older; able-bodied; no cognitive or visual impairments; fluent in spoken and written English; and not diagnosed with epilepsy. Participant ages ranged from 19 to 24 years old with torso heights varied between 0.51 and 0.58 meters.

For each participant, we conducted a total of 16 trials in VR with four trials for each of the four assistive tasks. The four trials for each task were organized such that we evaluated both the Original and Revised policies on both the PR2 and Jaco robots. We randomized the ordering of assistive tasks, robots, and control policies for each participant according to a randomized counterbalancing design.

During the study, participants wore the virtual reality headset and held both controllers. Participants sat in a wheelchair for the itch scratching, feeding, and drinking tasks and laid in a hospital bed for bed bathing assistance. Similar to the simulation environments used for training control policies, each trial in virtual reality lasted 20 seconds for a total of 200 time steps. The robot executed an action at each time step in virtual reality, once every 0.1 seconds. Participants were instructed that they could move their arms, upper body,

and head to interact with the simulated robots towards the goal of successfully receiving assistance for each task. Prior to each assistive task, we gave participants an unscripted practice trial to familiarize themselves on interacting with the robot and how to accomplish the task in virtual reality. We randomly selected either the PR2 or Jaco robot and used the Revised control policies (see Chapter 7.5.2) for each practice trial. In order to elicit a wide range of human motion and interactions throughout our study, we instructed participants to accomplish a task (e.g. drink water from the cup), but we did not provide instructions on how to interact with the robot or how to appropriately complete the task. Our study and experiments can be found in the supplementary video.

7.6.2 Objective Measures

We used the reward functions and success percentages defined in Assistive Gym as objective measures of performance [199].

- **Feeding**: The robot is rewarded for moving the spoon closer to the person's mouth and when food enters the person's mouth. The robot is penalized for dropping food or applying large forces to the person. Task success is defined by the robot feeding at least 75% of all food particles to the person's mouth.

- **Drinking Water**: The robot is rewarded for moving the cup closer to the person's mouth, tilting the cup for drinking, and when water particles enter the person's mouth. The robot is penalized for spilling water or applying large forces to the person. Task success is defined by the robot pouring at least 75% of all water into the person's mouth.

- **Itch Scratching**: The robot is rewarded for moving the scratching tool closer to the itch location and for performing scratching motions around the itch. The robot is penalized for applying large forces to the person, or more than 10 N of force [70] near the itch. Task success is defined by the robot performing at least 25 scratching motions near the itch.

- **Bed Bathing**: The robot is rewarded for moving the washcloth closer to the person's arm and for using the bottom of the washcloth to wipe off markers uniformly distributed (3 cm apart) along the person's right arm. The robot is penalized for applying large forces to the person. Task success is defined by the robot wiping off at least 30% of all markers along a person's arm)

7.6.3 Subjective Measures

In order to assess participants' perceptions of the control policies, we used a questionnaire with four statements pertaining to perceptions of the robot's performance, safety, comfort and speed. For each statement, participants were asked to record how much they agreed with the statement on an interval scale from 1 (strongly disagree) to 7 (strongly agree) with 4 being neutral. We based this 7-point scale questionnaire on past work on robot-assisted feeding [220].

The four statements follow:

- L1: The robot successfully assisted me with the task.
- L2: I felt comfortable with how the robot assisted me in VR.
- L3: I felt I would be safe if this were a real robot.
- L4: The robot moved with appropriate speed.

7.7 Results

7.7.1 Performance of the Original Policies

Table 7.1 depicts the average reward and task success when both robots used the Original policies in simulation with static human models and in AVR Gym with the eight human participants. When evaluating the Original policy for a given robot and assistive task in virtual reality, we averaged rewards and task success over trials from all eight participants.

Table 7.1: Average reward and task success between simulation and virtual reality using the Original control policies.

	Simulation		Virtual Reality	
Task	PR2	Jaco	PR2	Jaco
Feeding	103 (77%)	82 (75%)	122 (100%)	37 (63%)
Drinking	380 (59%)	144 (32%)	502 (75%)	-92 (0%)
Scratching	62 (35%)	218 (48%)	25 (25%)	-69 (0%)
Bathing	85 (13%)	118 (28%)	-87 (0%)	-126 (0%)
Avg. Success	46%	46%	50%	16%

For simulation, we averaged rewards and task success over 100 simulation rollouts with a random initial human pose for each rollout. Since the simulated human holds a static pose throughout an entire trial, we take an average over a larger number of simulation trials to evaluate performance over multiple human poses. We note that each task uses a slightly different reward function due to task-specific reward elements and hence reward values are not directly comparable across different tasks.

For the feeding assistance task, the Original policies achieved similar average reward and task success with real people as they achieved in simulation with fixed human models. However, performance was more varied for the drinking, itch scratching, and bed bathing tasks. The most noticeable errors occurred with the itch scratching and bed bathing tasks, where the Original policies for both the Jaco and PR2 robots frequently failed to move their end effectors closer to a person's body. These errors can be visually seen in Figure 7.4 where the PR2 actively moved away from the person during bed bathing assistance and in Figure 7.5 where the Jaco robot actuated itself to behind the wheelchair rather than scratching an itch on the person's arm. Differences in results between the two robots can be partially attributed to differences in robot kinematics and base positioning. The Jaco is mounted to a fixed position on the wheelchair, whereas we optimize the PR2's base pose near a person according to joint-limit-weighted kinematic isotropy (JLWKI) and task-centric manipulability metrics [199, 174].

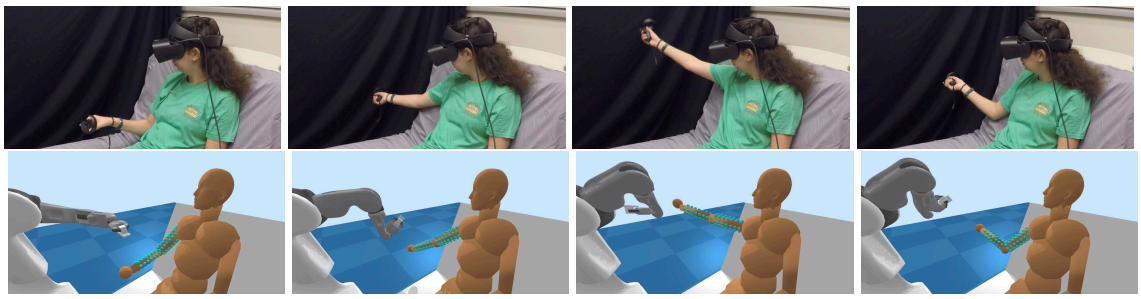


Figure 7.4: The PR2 robot fails to provide assistance with bed bathing in virtual reality when using the Original control policy. Blue markers are placed uniformly around the person's right arm, which the robot can clean off with the bottom of the wiping tool.

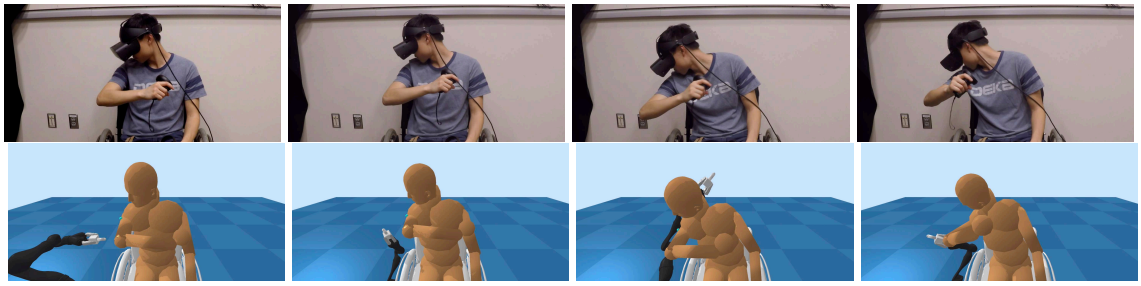


Figure 7.5: The Jaco fails to provide itch scratching assistance in virtual reality with the Original control policy. The itch is represented by a blue marker.

Except for responses about the robot's speed (L4), participants' average responses were negative or neutral (L1, L2, and L3) (see Figure 7.6). Most notably, participants tended to perceive the robots as being unsuccessful at assistance, as evidenced by responses to L1 being significantly below neutral (4) with $p < 0.05$ using a Wilcoxon signed-rank test.

Responses did vary by task. When the robots used the Original policies for itch scratching assistance, participants slightly disagreed with the statement on successful assistance (L1), with an average response of 3.3 across all participants. When using the Original policies for bed bathing assistance, participants reported an even lower average rating of 1.2, where almost all participants strongly disagreed that the robots provided successful bathing assistance.

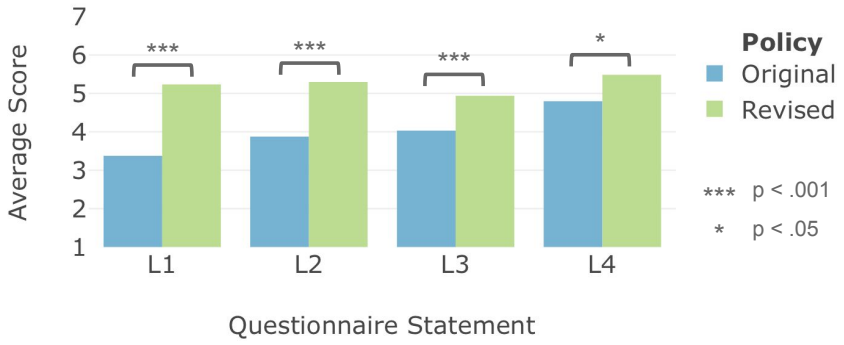


Figure 7.6: Comparison of 7-point scale questionnaire responses for both Original and Revised policies. Responses are averaged over all participants, tasks, and robots. p-values are computed with a Wilcoxon signed-rank test.

Table 7.2: Average reward and task success for the Original and Revised control policies in VR with real people.

	Original Policies		Revised Policies	
Task	PR2	Jaco	PR2	Jaco
Feeding	122 (100%)	37 (63%)	113 (100%)	36 (45%)
Drinking	502 (75%)	-92 (0%)	458 (75%)	199 (42%)
Scratching	25 (25%)	-69 (0%)	36 (45%)	95 (62%)
Bathing	-87 (0%)	-126 (0%)	-18 (0%)	-16 (0%)
Avg. Success	50%	16%	**55%**	**37%**

7.7.2 Performance of the Revised Policies

Table 7.2 compares the average reward and task success between the Original and Revised control policies during our VR study. For a given task and robot, we average the rewards and task success rates over trials from all eight participants.

From the table, we observe that the Revised policies for both robots outperformed the Original policies when providing itch scratching and bed bathing assistance. Figure 7.7 demonstrates the Jaco robot using a Revised policy to provide bed bathing assistance to a participant in virtual reality. While the Original policies for the Jaco robot failed to reliably assist human participants with drinking, we observe that the Revised policies for the Jaco exhibited a greater level of success. Figure 7.8 depicts an image sequence of

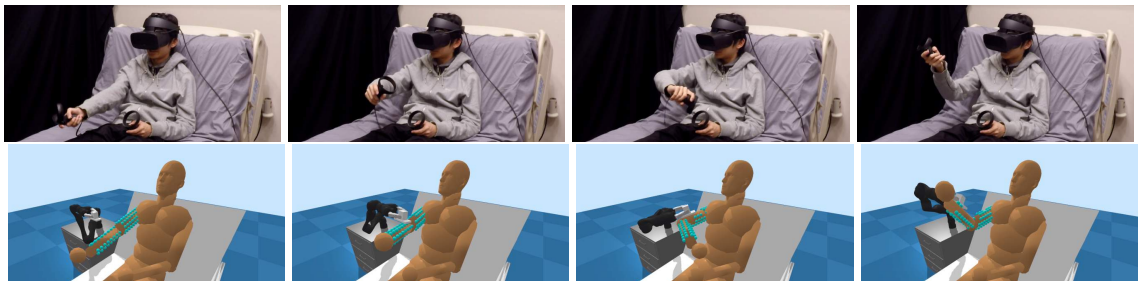


Figure 7.7: Image sequence of a table-mounted Jaco providing bed bathing assistance using the Revised control policy.

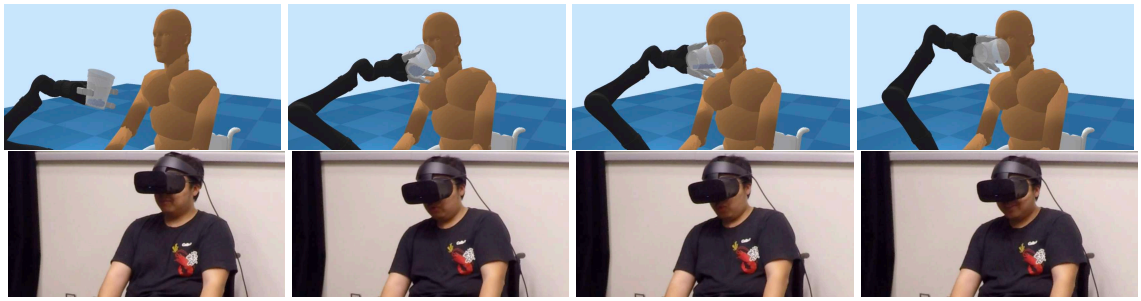


Figure 7.8: Image sequence of drinking assistance with the Jaco and the Revised control policy.

the Jaco robot using a Revised control policy to provide drinking assistance. Based on these success metrics and qualitative image sequences, we find that the Revised polices exhibited reasonable performance for all of the assistive tasks. However, we note that these revised controllers are not perfect and still exhibit some errors. This is especially true for hard to reach locations, such as scratching an itch underneath a person's arm. These control policies also assume that the person receiving assistance acts collaboratively with the robot. Additional research is required to explore how robot controllers learned in simulation can act appropriately when exposed to uncooperative or adversarial human motion.

For all four statements (L1, L2, L3, and L4), participants' average responses were positive or neutral (see Figure 7.6). Most notably, participants tended to perceive the robots as being successful at assistance, as evidenced by responses to L1 being significantly above neutral (4) with $p < 0.001$ using a Wilcoxon signed-rank test.

Figure 7.6 displays a comparison of questionnaire responses between the Original and Revised policies when responses are averaged over all participants, tasks, and robots for

each questionnaire statement. On average, participants provide higher responses for trials in which a robot used a Revised policy to provide assistance. To determine a statistical difference between the Original and Revised policies, we apply a Wilcoxon signed-rank test, a non-parametric test, between participant responses from trials using the Original control policies and responses when using the Revised policies. The computed p-values are depicted in Figure 7.6, where we observed a statistically significant difference at the $p < .001$ level for the first three questions relating to task success, comfort, and safety. These results indicate that participants perceived a noticeable improvement in a robot's performance and safety when using Revised control policies that were trained based on biomechanics feedback from virtual reality.

Overall, we found that that our Revised policies significantly outperformed the Original policies, which indicates that VR can be used to efficiently improve simulation-trained controllers for physical human-robot interaction.

7.7.3 Posthoc Biomechanical Analysis

Our iterative research and development using AVR Gym, discussed in Chapter 7.5.2, resulted in Revised policies trained with new distributions of human torso heights and waist orientations. Here, we provide a posthoc statistical analysis of these biomechanical parameters for our simulated humans and our real participants in virtual reality.

During our human study, we aligned the human model in virtual reality with a real participant's pose, as described in Chapter 7.4. This alignment enabled us to record the full state, height, and pose of a participant at each time step during the virtual reality trials.

To determine whether a statistical difference exists between human torso heights, we applied a Wilcoxon signed-rank test between human torso heights in the simulation environments used to train the Original policies and torso heights of human participants in virtual reality. This test returned a p-value of $p < 0.001$, indicating that the original simulation environments did not properly model the distribution of real human heights observed

during virtual reality.

We then applied the same test to a set of torso heights from the updated simulation environments used to train the Revised policies. When compared to torso heights of human participants, the test returns a p-value of $p = 0.29$, which indicates that there is a less significant difference between the distributions of human heights used to train the Revised control policies and human heights observed in virtual reality.

In order to evaluate differences in waist orientations between simulation and virtual reality, we consider the 3D position of a person's head, averaged over time. Since the human model in virtual reality is aligned with a real participant's pose, we track the 3D head position of a participant by querying the 3D head position of the virtual human model. In order to separate biomechanical differences in torso height from waist orientations, we modified the original simulation environments to include random variation among human torso heights, but no variation among waist orientations. Using a Wilcoxon signed-rank test, we found a statistically significant difference in head position (averaged over all time steps for a trial) between data from these new simulation environments and the head positions of participants in virtual reality, with a p-value of $p < 0.01$. However, by performing this same test between head positions from the revised simulation environments (Chapter 7.5.2) and from VR, we found a p-value of $p = 0.26$, indicating that the improved simulation environments used to train our Revised policies better matched the biomechanics of our participants in VR.

7.8 Conclusion

We presented Assistive VR Gym (AVR Gym), which enables real people to interact with virtual assistive robots. We also provided evidence that AVR Gym can help researchers improve the performance of simulation-trained assistive robots. Overall, our results suggest that VR can be used to help bridge the gap between physics simulations and the real world by enabling real people to interact with virtual robots.

CHAPTER 8

TAKEAWAYS AND NEXT STEPS

8.1 Physics-based Simulation for Assistive Robotics

We have introduced Assistive Gym, the first open source physics simulation framework for physically assistive robots. Physics simulation can open up entirely new research directions and opportunities within physical human-robot interaction. For example, our research has begun investigating how we can use this tool to develop versatile intelligent robotic caregivers. We have introduced several physics-based environments for physical assistance—each designed around training assistive robots and developing control algorithms towards common sense reasoning.

Physics simulation has enabled us and other researchers to develop baselines and benchmarks for robotic caregiving. We have also begun investigating generalization of physical human-robot interaction from a number of angles, including human motion, preferences, and variation in human body shape and impairments. There are many directions in which robots may leverage simulation in order to improve how they interact with or understand the human body. We are making progress, but there remains significant work remaining in each of the research directions outlined above.

One substantial question remains: "Do assistive robot controllers designed in physics simulation transfer to real robot hardware interacting with real people?". This ties into the need for extensive real-world evaluation of assistive robot controllers designed in Assistive Gym. This also relates to notions of sim-to-real transfer of controllers. There are several existing techniques that may be of benefit towards real-world evaluation, including transfer learning, system identification, domain randomization, and fine-tuning simulation-trained controllers on real-world data. We have begun to investigate transferring simulation-trained

policies to real mobile manipulators, as outlined in the following section, yet this remains one of the most substantial and promising directions for future research.

Pushing beyond the technical challenges of sim-to-real transfer, the core purpose of physically assistive robotics is to provide real benefit to the target population: adults with physical disabilities who require assistance with activities of daily living. It is crucial to keep this target population involved throughout the research process and to evaluate designed robotic solutions with this population.

Although physics engines are widely available for both the entertainment and academic industries, there remains significant room for development and improvement of physics simulators for robotics. This is especially the case for simulating realistic human biomechanics or realistic interactions between robots and the human body. Although there are many physics engines capable of computing realistic rigid-body dynamics, there are fewer options for engines capable of soft-body dynamics, and the options are even more constrained for engines that handle coupled rigid- and soft-body dynamics. While many mobile manipulators can be modeled as a rigid multibody, the human body should ideally be modeled as a deformable object in simulation and preferably as a non-uniform deformable. Many objects in human environments are also highly deformable, including blankets, clothes, beds, sponges, and towels. Soft-body simulation environments that are both accurate and fast enough to be used in real-world assistive robotics contexts remain as a largely open research direction. Liu and Negrut [221] provide a comprehensive overview of the current state-of-the-art for physics-based simulators in robotics.

8.2 Safely Transferring from Simulation to Reality

Ensuring human safety is critical during physical human-robot interaction, especially in scenarios when robots assist people with impairments. Transferring robot controllers from simulation to real hardware has been demonstrated for topics such as manipulation [151, 152, 153] or quadruped locomotion [150, 222], yet these past approaches rely on robots

operating in isolated environments away from people. Our prior research has begun to investigate transferring from simulation to reality for robot-assisted dressing [70, 133, 169], yet these past approaches have been heavily specialized for dressing assistance and have not focused on how to ensure people receiving assistance from the robot remain safe. It remains an open problem how to ensure simulation-trained controllers are safe when tested with human researchers, or evaluated with human participants. There are a number of approaches which may benefit the transfer of assistive robot controllers to real hardware, while ensuring that these controllers remain safe when interacting with real people.

One direction is to further develop methods to safely evaluate simulation-trained control policies in the real world, without the need for physical human-robot interaction. This may include using both virtual reality (real people interacting with virtual robots, see Chapter 7.4) and 'robot augmented reality' (real robots interacting with virtual people). This may include developing a set of metrics (task success, reward, robot kinematics, etc.) to evaluate simulation-trained controllers performance in the real world before interactions with real people. Robot augmented reality works by having a real robot provide assistance to a virtual human by means of a physics simulation (i.e. Assistive Gym) that models an actuated virtual human and is aligned with the real robot's kinematic state.

We want a robot and its (simulation-trained) controller to be safe when interacting with a person for the very first time (often a researcher/developer), and then every time after that (participants with/without disabilities). As a step towards ensuring human safety during physical robotic assistance, one option is to first capture a personalized 3D human model for the person receiving assistance and iteratively fine tune the robot's existing policy so that it is specialized to that person's physique. A customized 3D human model can be attained either from 2D images [223, 224] or from pressure sensing on a bed [132, 67, 225]. Additional safety measures may include execution monitoring (inspired by Park et al. [46, 163, 220]) with fixed thresholds over multiple sensory modalities during assistance, including end effector force/torque sensing, visually estimated human body velocities, and

robot kinematics.

8.3 Capacitive Servoing and Haptics for Physical Interaction

Another promising direction to advance assistive robotics is the development of new multimodal sensors and analysis techniques for robots to more intelligently sense and interact with the human body. This research direction is based off the notion that the sense of touch is an extremely rich modality that presents an opportunity to greatly improve how robots interact with the human body. So far, we have shown how a robot can use haptic and capacitive measurements to improve physical interactions with the human body. We have shown how a robot can use haptic measurements from its end effector to perform haptic perspective-taking by predicting how future actions would apply forces onto a person's body during robot-assisted dressing (Chapter 2). We have also demonstrated how multidimensional capacitive servoing can enable a mobile manipulator to navigate its end effector along the contours of human limbs (Chapter 5). Techniques for simulating capacitance signals around the human body may also serve as a promising approach to improve capacitive servoing methods in future work.

Although this past work has made progress towards sensing techniques for intelligent physical interactions and assistance, there also remains gaps in a robot's abilities to leverage multimodal haptic data near people. As a step towards multimodal haptic sensors, we have investigated multimodal force, vibration, and active temperature sensing, coupled with semi-supervised learning for material recognition (metal, plastic, wood, fabric, etc.) of household objects [226]. More recently, we have explored how robots can use near-infrared spectroscopy with micro spectrometers for object material recognition. We have found that the local measurements from a handheld spectrometer can result in highly accurate material recognition performance even with small amounts of training data [227]. We have then coupled spectroscopy with high resolution texture imaging, which has allowed us to investigate pixel-level recognition of common household materials [228].

Looking forward, I intend to investigate how robots can leverage all of this local sensory information to inform their interactions with people. There are several potential paths for future work, including haptic sensing for safe touch along the human body, using sensing to localize contact along the body, and simulation of haptic sensing with people.

Appendices

APPENDIX A

CAPACITIVE SERVOING DATA COLLECTION

To collect a dataset of capacitance measurements for training a pose estimator, we first position the robot's end effector and capacitive sensor above the participant's limb (Data Collection 1, line 2) with an initial relative pose of $\boldsymbol{p}_0 = (0, 0, 0, 0)$. The robot begins by selecting a target end effector pose above the limb (line 6), $\boldsymbol{p}_T = (D_{T,y}, D_{T,z}, \theta_{T,y}, \theta_{T,z})$. In our implementation, both the position and orientation are selected from a uniform distribution such that $D_{T,y} \in [-10 \text{ cm}, 10 \text{ cm}]$, $D_{T,z} \in [0 \text{ cm}, 15 \text{ cm}]$, and $\theta_{T,y}, \theta_{T,z} \in [-\frac{\pi}{8}, \frac{\pi}{8}]$. We denote this bounded space of end effector poses above the person's static limb as S, which is depicted in Fig. 5.9. The robot then selects target translational and rotational velocities for its end effector (lines 7-8) as it moves to the target pose $\boldsymbol{p}_T$. We selected translational velocities $\boldsymbol{v}_D$ and rotational velocities $\boldsymbol{v}_\theta$ from a uniform distribution such that $||\boldsymbol{v}_D||_2 \in [3 \text{ cm/s}, 10 \text{ cm/s}]$ and $||\boldsymbol{v}_\theta||_2 \in [\frac{\pi}{20}, \frac{\pi}{8}]$ in radian/s.

We then compute the per time step change in the robot end effector's Cartesian position $\Delta\boldsymbol{\omega}$ and Euler orientation $\Delta\boldsymbol{\phi}$ (lines 9-13). $\Delta\boldsymbol{\omega}$ and $\Delta\boldsymbol{\phi}$ are defined such that the end effector progresses to the target pose $\boldsymbol{p}_T$ with velocities $\boldsymbol{v}_D$ and $\boldsymbol{v}_\theta$ (lines 12-13), where $\odot$ defines the Hadamard product.

For each time step t during data collection, the robot records a measurement from all capacitive sensor electrodes $\boldsymbol{c}_t$ (line 17) and the current pose difference between the sensor and human limb $\boldsymbol{p}_t$, which is calculated via forward kinematics (lines 18-21). Using inverse kinematics, the robot then executes a single step action (lines 23-30) to move the capacitive sensor towards the target state $\boldsymbol{p}_T$ along a linear trajectory with velocities $\boldsymbol{v}_D$ and $\boldsymbol{v}_\theta$. Once the robot reaches the target state $\boldsymbol{p}_t \approx \boldsymbol{p}_T$, we store all recorded data from the trajectory into a dataset $\mathcal{D}$ (line 32) and have the robot progress to the next randomly generated target pose for a total of $N = 500$ trajectories (line 5). This data collection process is then repeated for

three locations above the arm (wrist, forearm, upper arm) and three locations above the leg (ankle, shin, knee), which resulted in a total of $N = 3000$ end effector trajectories used to collect capacitance data around a participant's limbs.

Data Collection 1 Capacitive Sensor Data Collection for Pose Estimation

Given: N: number of trajectories,
$\boldsymbol{p}_0$: initial pose,
S: space of end effector poses,
τ: data collection frequency.
Move capacitive sensor to top of limb; to $\boldsymbol{p}_0$.
$\boldsymbol{\omega}_0, \boldsymbol{\phi}_0 \leftarrow$ GetCurrentEndEffectorPose().
$\mathcal{D} \leftarrow \{\}$
for $i = 1, \ldots, N$ **do**
    Select target $\boldsymbol{p}_T \leftarrow (D_{T,y}, D_{T,z}, \theta_{T,y}, \theta_{T,z}) \in S$.
    Select translational velocities $\boldsymbol{v}_D \leftarrow (0, v_{D_y}, v_{D_z})$.
    Select rotational velocities $\boldsymbol{v}_\theta \leftarrow (0, v_{\theta_y}, v_{\theta_z})$.
    $\boldsymbol{\omega}_I, \boldsymbol{\phi}_I \leftarrow$ GetCurrentEndEffectorPose().
    $\boldsymbol{\omega}_T \leftarrow \boldsymbol{\omega}_0 + (0, D_{T,y}, D_{T,z})$.
    $\boldsymbol{\phi}_T \leftarrow \boldsymbol{\phi}_0 + (0, \theta_{T,y}, \theta_{T,z})$.
    $\Delta\boldsymbol{\omega} \leftarrow (\boldsymbol{\omega}_T - \boldsymbol{\omega}_I) \odot \frac{\boldsymbol{v}_D}{\tau}$.
    $\Delta\boldsymbol{\phi} \leftarrow (\boldsymbol{\phi}_T - \boldsymbol{\phi}_I) \odot \frac{\boldsymbol{v}_\theta}{\tau}$.
    $\delta \leftarrow \{\}$.
    $t \leftarrow 1$.
    while $\boldsymbol{p}_{t-1} \not\approx \boldsymbol{p}_T$ **do**
        $\boldsymbol{c}_t \leftarrow$ GetCapacitanceMeasurements().
        $\boldsymbol{\omega}_t, \boldsymbol{\phi}_t \leftarrow$ GetCurrentEndEffectorPose().
        $(D_{t,x}, D_{t,y}, D_{t,z}) \leftarrow \boldsymbol{\omega}_t - \boldsymbol{\omega}_0$.
        $(\theta_{t,x}, \theta_{t,y}, \theta_{t,z}) \leftarrow \boldsymbol{\phi}_t - \boldsymbol{\phi}_0$.
        $\boldsymbol{p}_t \leftarrow (D_{t,y}, D_{t,z}, \theta_{t,y}, \theta_{t,z})$.
        $\delta \leftarrow \delta \cup \{(\boldsymbol{c}_t, \boldsymbol{p}_t)\}$.
        $\boldsymbol{\omega}_t^* \leftarrow \boldsymbol{\omega}_I + t\Delta\boldsymbol{\omega}$
        $\boldsymbol{\phi}_t^* \leftarrow \boldsymbol{\phi}_I + t\Delta\boldsymbol{\phi}$
        if $\boldsymbol{\omega}_t \approx \boldsymbol{\omega}_T$ **then**
            $\boldsymbol{\omega}_t^* \leftarrow \boldsymbol{\omega}_T$
        else if $\boldsymbol{\phi}_t \approx \boldsymbol{\phi}_T$ **then**
            $\boldsymbol{\phi}_t^* \leftarrow \boldsymbol{\phi}_T$
        $\boldsymbol{\alpha}_T \leftarrow$ InverseKinematics($\boldsymbol{\omega}_t^*, \boldsymbol{\phi}_t^*$).
        SendToActuators($\boldsymbol{\alpha}_t$).
        $t \leftarrow t + 1$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\delta\}$
return dataset $\mathcal{D}$

REFERENCES

[1] J. M. Wiener, R. J. Hanley, R. Clark, and J. F. Van Nostrand, "Measuring the activities of daily living: Comparisons across national surveys," *Journal of gerontology*, vol. 45, no. 6, S229–S237, 1990.

[2] P. van der Smagt, M. A. Arbib, and G. Metta, "Neurorobotics: From vision to action," in *Springer handbook of robotics*, Springer, 2016, pp. 2069–2094.

[3] N. Koganti, T. Tamei, T. Matsubara, and T. Shibata, "Estimation of human cloth topological relationship using depth sensor for robotic clothing assistance," in *Proceedings of Conference on Advances In Robotics*, ACM, 2013, pp. 1–6.

[4] ——, "Real-time estimation of human-cloth topological relationship using depth sensor for robotic clothing assistance," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2014, pp. 124–129.

[5] N. Koganti, J. G. Ngeo, T. Tomoya, K. Ikeda, and T. Shibata, "Cloth dynamics modeling in latent spaces and its application to robotic clothing assistance," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 3464–3469.

[6] L. Twardon and H. Ritter, "Interaction skills for a coat-check robot: Identifying and handling the boundary components of clothes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 3682–3688.

[7] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, "Reinforcement learning of clothing assistance with a dual-arm robot," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, IEEE, 2011, pp. 733–738.

[8] S. D. Klee, B. Q. Ferreira, R. Silva, J. P. Costeira, F. S. Melo, and M. Veloso, "Personalized assistance for dressing users," in *International Conference on Social Robotics*, Springer, 2015, pp. 359–369.

[9] Y. Gao, H. J. Chang, and Y. Demiris, "User modelling for personalised dressing assistance by humanoid robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 1840–1845.

[10] K. Yamazaki, R. Oya, K. Nagahama, K. Okada, and M. Inaba, "Bottom dressing by a life-sized humanoid robot provided failure detection and recovery functions," in *System Integration (SII), 2014 IEEE/SICE International Symposium on*, IEEE, 2014, pp. 564–570.

[11] Y. Gao, H. J. Chang, and Y. Demiris, "Iterative path optimisation for personalised dressing assistance using vision and force information," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 4398–4403.

[12] A. Kapusta, W. Yu, T. Bhattacharjee, C. K. Liu, G. Turk, and C. C. Kemp, "Data-driven haptic perception for robot-assisted dressing," in *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*, IEEE, 2016, pp. 451–458.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[16] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[17] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 6645–6649.

[18] A. I. Aviles, S. Alsaleh, P. Sobrevilla, and A. Casals, "Exploring the effects of dimensionality reduction in deep networks for force estimation in robotic-assisted surgery," in *SPIE Medical Imaging*, International Society for Optics and Photonics, 2016.

[19] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 4525–4532.

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *Aistats*, vol. 9, 2010, pp. 249–256.

[21] W. Yu, A. Kapusta, J. Tan, C. C. Kemp, G. Turk, and C. K. Liu, "Haptic data simulation for robot-assisted dressing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017.

[22] N. Hansen, "The CMA evolution strategy: A tutorial," *Technische Universitat Berlin, TU Berlin*, 2016.

[23] F. Chollet, *Keras*, https://github.com/fchollet/keras, 2015.

[24] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, 2012.

[25] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2786–2793.

[26] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in Clutter with Whole-Arm Tactile Sensing," *International Journal of Robotics Research*, vol. 32, no. 4, pp. 458–482, 2013.

[27] I. Lenz, R. A. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control.," *Robotics: Science and Systems*, 2015.

[28] N. Koganti, T. Tamei, K. Ikeda, and T. Shibata, "Bayesian nonparametric learning of cloth models for real-time state estimation," *IEEE Transactions on Robotics*, 2017.

[29] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.

[30] Z. Erickson, A. Clegg, W. Yu, G. Turk, C. K. Liu, and C. C. Kemp, "What does the person feel? learning to infer applied forces during robot-assisted dressing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 6058–6065.

[31] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning.," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[32] J. Bellingham, A. Richards, and J. How, "Receding horizon control of autonomous aerial vehicles," in *American Control Conference*, 2002.

[33] T. Erez, Y. Tassa, and E. Todorov, "Infinite-horizon model predictive control for periodic tasks with contacts.," *Robotics: Science and systems VII*, vol. 73, 2012.

[34] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 137–142.

[35] K. Chow and C. C. Kemp, "Robotic repositioning of human limbs via model predictive control.," in *2016 IEEE International Symposium on Robot and Human Interactive Communication*, 2016, pp. 473–480.

[36] M. Dominici and R. Cortesao, "Model predictive control architectures with force feedback for robotic-assisted beating heart surgery.," in *In Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 2276–2282.

[37] V. Duchaine, S. Bouchard, and C. Gosselin, "Computationally efficient predictive robot control," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 5, pp. 570–578, 2007.

[38] J. Fu, S. Levine, and P. Abbeel, "One-shot learning of manipulation skills with online dynamics adaptation and neural network priors," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 4019–4026.

[39] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: A survey.," *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[40] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *Advances in neural information processing systems*, 2015, pp. 2746–2754.

[41] B. Boots, A. Byravan, and D. Fox, "Learning predictive models of a depth camera & manipulator from raw execution traces," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 4021–4028.

[42] G. Chance, A. Camilleri, B. Winstone, P. Caleb-Solly, and S. Dogramadzi, "An assistive robot to support dressing-strategies for planning and error handling," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, 2016, pp. 774–780.

[43] K. Yamazaki, R. Oya, K. Nagahama, and M. Inaba, "A method of state recognition of dressing clothes based on dynamic state matching," in *System Integration (SII), 2013 IEEE/SICE International Symposium on*, IEEE, 2013, pp. 406–411.

[44] P. Jiménez, "Visual grasp point localization, classification and state recognition in robotic manipulation of cloth: An overview," *Robotics and Autonomous Systems*, vol. 92, pp. 107–125, 2017.

[45] D. Shinohara, T. Matsubara, and M. Kidode, "Learning motor skills with non-rigid materials by reinforcement learning," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2011, pp. 2676–2681.

[46] D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta, and C. C. Kemp, "A multimodal execution monitor with anomaly classification for robot-assisted feeding," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 5406–5413.

[47] G. Chance, A. Jevtić, P. Caleb-Solly, and S. Dogramadzi, "A quantitative analysis of dressing dynamics for robotic dressing assistance," *Frontiers in Robotics and AI*, vol. 4, 2017.

[48] A. Kapusta, W. Yu, T. Bhattacharjee, C. K. Liu, G. Turk, and C. C. Kemp, "Data-driven haptic perception for robot-assisted dressing," in *2016 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 2016, pp. 451–458.

[49] H. Muhammad, C. Recchiuto, C. Oddo, L. Beccai, C. Anthony, M. Adams, M. Carrozza, and M. Ward, "A capacitive tactile sensor array for surface texture discrimination," *Microelectronic Engineering*, vol. 88, no. 8, pp. 1811–1813, 2011.

[50] A. Damilano, P. M. Ros, A. Sanginario, A. Chiolerio, S. Bocchini, I. Roppolo, C. F. Pirri, S. Carrara, D. Demarchi, and M. Crepaldi, "A robust capacitive digital read-out circuit for a scalable tactile skin," *IEEE Sensors Journal*, vol. 17, no. 9, pp. 2682–2695, 2017.

[51] D. P. Cotton, I. M. Graz, and S. P. Lacour, "A multifunctional capacitive sensor for stretchable electronic skins," *IEEE Sensors Journal*, vol. 9, no. 12, pp. 2008–2009, 2009.

[52] J. Ulmen and M. Cutkosky, "A robust, low-cost and low-noise artificial skin for human-friendly robots," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2010, pp. 4836–4841.

[53] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 389–400, 2011.

[54] S. Phan, Z. F. Quek, P. Shah, D. Shin, Z. Ahmed, O. Khatib, and M. Cutkosky, "Capacitive skin sensors for robot impact monitoring," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2011, pp. 2992–2997.

[55] Z. Ji, H. Zhu, H. Liu, N. Liu, T. Chen, Z. Yang, and L. Sun, "The design and characterization of a flexible tactile sensing array for robot skin," *Sensors*, vol. 16, no. 12, p. 2001, 2016.

[56] F. Xia, B. Bahreyni, and F. Campi, "Multi-functional capacitive proximity sensing system for industrial safety applications," in *2016 IEEE Sensors*, IEEE, 2016, pp. 1–3.

[57] T. Schlegl, T. Kröger, A. Gaschler, O. Khatib, and H. Zangl, "Virtual whiskers—highly responsive robot collision avoidance," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 5373–5379.

[58] A. Hoffmann, A. Poeppel, A. Schierl, and W. Reif, "Environment-aware proximity detection with capacitive sensors for human-robot-interaction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 145–150.

[59] N. Kirchner, D. Hordern, D. Liu, and G. Dissanayake, "Capacitive sensor for object ranging and material type identification," *Sensors and Actuators A: Physical*, vol. 148, no. 1, pp. 96–104, 2008.

[60] H.-K. Lee, S.-I. Chang, and E. Yoon, "Dual-mode capacitive proximity sensor for robot application: Implementation of tactile and proximity sensing capability on a single polymer platform using shared electrodes," *IEEE Sensors Journal*, vol. 9, no. 12, pp. 1748–1755, 2009.

[61] S. E. Navarro, M. Marufo, Y. Ding, S. Puls, D. Göger, B. Hein, and H. Wörn, "Methods for safe human-robot-interaction using capacitive tactile proximity sensors," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 1149–1154.

[62] S. E. Navarro, M. Schonert, B. Hein, and H. Wörn, "6d proximity servoing for preshaping and haptic exploration using capacitive tactile proximity sensors," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2014, pp. 7–14.

[63] J. Vespa, D. M. Armstrong, and L. Medina, "Demographic turning points for the united states: Population projections for 2020 to 2060," *Washington, DC: US Census Bureau*, vol. 42, 2018.

[64] T. L. Mitzner, T. L. Chen, C. C. Kemp, and W. A. Rogers, "Identifying the potential for robotics to assist older adults in different living environments," *International journal of social robotics*, vol. 6, no. 2, pp. 213–227, 2014.

[65] D. Goeger, M. Blankertz, and H. Woern, "A tactile proximity sensor," *IEEE Sensors*, vol. 143, no. 2, pp. 589–594, 2010.

[66] Z. Erickson, M. Collier, A. Kapusta, and C. C. Kemp, "Tracking human pose during robot-assisted dressing using single-axis capacitive proximity sensing," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2245–2252, 2018.

[67] H. M. Clever, A. Kapusta, D. Park, Z. Erickson, Y. Chitalia, and C. C. Kemp, "3d human pose estimation on a configurable bed from a pressure image," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 54–61.

[68] G. Chance, A. Jevtić, P. Caleb-Solly, G. Alenya, C. Torras, and S. Dogramadzi, ""elbows out"—predictive tracking of partially occluded pose for robot-assisted dressing," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3598–3605, 2018.

[69] F. Zhang, A. Cully, and Y. Demiris, "Personalized robot-assisted dressing using user modeling in latent spaces," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 3603–3610.

[70] Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, and C. C. Kemp, "Deep haptic model predictive control for robot-assisted dressing," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 4437–4444.

[71] H. Satoh, T. Kawabata, and Y. Sankai, "Bathing care assistance with robot suit hal," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp. 498–503.

[72] K. Bezerra, J. Machado, V. Carvalho, M. Castro, P. Costa, D. Matos, and F. Soares, "Bath-ambience–a mechatronic system for assisting the caregivers of bedridden people," *Sensors*, vol. 17, no. 5, p. 1156, 2017.

[73] A. K. Carruth, D. Ricks, and P. Pullen, "Bag baths: An alternative to the bed bath," *Nursing Management*, vol. 26, no. 9, p. 75, 1995.

[74] A. Zlatintsi, I. Rodomagoulakis, P. Koutras, A. Dometios, V. Pitsikalis, C. S. Tzafestas, and P. Maragos, "Multimodal signal processing and learning aspects of human-robot interaction for an assistive bathing robot," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 3171–3175.

[75] C.-H. King, T. L. Chen, A. Jain, and C. C. Kemp, "Towards an assistive robot that autonomously performs bed baths for patient hygiene," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, IEEE, 2010, pp. 319–324.

[76] C. A. Okoro, N. D. Hollis, A. C. Cyrus, and S. Griffin-Blake, "Prevalence of disabilities and health care access by disability status and type among adults—united states, 2016," *Morbidity and Mortality Weekly Report*, vol. 67, no. 32, p. 882, 2018.

[77] Z. Erickson, H. M. Clever, V. Gangaram, G. Turk, C. K. Liu, and C. C. Kemp, "Multidimensional capacitive sensing for robot-assisted dressing and bathing," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, IEEE, 2019, pp. 224–231.

[78] S. Lee, W. Buxton, and K. Smith, "A multi-touch three dimensional touch-sensitive tablet," *Acm Sigchi Bulletin*, vol. 16, no. 4, pp. 21–25, 1985.

[79] G. Barrett and R. Omote, "Projected-capacitive touch technology," *Information Display*, vol. 26, no. 3, pp. 16–21, 2010.

[80] J. Lee, M. Sinclair, M. Gonzalez-Franco, E. Ofek, and C. Holz, "Torc: A virtual reality controller for in-hand high-dexterity finger interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.

[81] J. A. Higgins, B. E. T. Rogoza, and S. S. Talati, *Hand-held controllers with capacitive touch sensors for virtual-reality systems*, US Patent 9,999,833, 2018.

[82] J. Cheng, O. Amft, and P. Lukowicz, "Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition," in *International conference on pervasive computing*, Springer, 2010, pp. 319–336.

[83] P. Holleis, A. Schmidt, S. Paasovaara, A. Puikkonen, and J. Häkkilä, "Evaluating capacitive touch input on clothes," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 2008, pp. 81–90.

[84] T. Grosse-Puppendahl, C. Holz, G. Cohn, R. Wimmer, O. Bechtold, S. Hodges, M. S. Reynolds, and J. R. Smith, "Finding common ground: A survey of capacitive sensing in human-computer interaction," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 3293–3315.

[85] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, "Applying electric field sensing to human-computer interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pp. 280–287.

[86] Y. Ye, C. Zhang, C. He, X. Wang, J. Huang, and J. Deng, "A review on applications of capacitive displacement sensing for capacitive proximity sensor," *IEEE Access*, vol. 8, pp. 45 325–45 342, 2020.

[87] S. Escaida Navarro, S. Mühlbacher-Karrer, H. Alagi, H. Zangl, K. Koyama, B. Hein, C. Duriez, and J. R. Smith, "Proximity Perception in Human-centered Robotics: A Survey on Sensing Systems and Applications," working paper or preprint, Mar. 2021.

[88] S. Tsuji and T. Kohama, "Proximity skin sensor using time-of-flight sensor for human collaborative robot," *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5859–5864, 2019.

[89] Y. Ding and U. Thomas, "Collision avoidance with proximity servoing for redundant serial robot manipulators," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 249–10 255.

[90] A. Poeppel, A. Hoffmann, M. Siehler, and W. Reif, "Robust distance estimation of capacitive proximity sensors in hri using neural networks," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2020, pp. 344–351.

[91] E. Zheng and Q. Wang, "Noncontact capacitive sensing-based locomotion transition recognition for amputees with robotic transtibial prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 2, pp. 161–170, 2016.

[92] S. Crea, S. Manca, A. Parri, E. Zheng, J. Mai, R. M. Lova, N. Vitiello, and Q. Wang, "Controlling a robotic hip exoskeleton with noncontact capacitive sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2227–2235, 2019.

[93] F. Xia, F. Campi, and B. Bahreyni, "Tri-mode capacitive proximity detection towards improved safety in industrial robotics," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5058–5066, 2018.

[94] C. Stetco, S. Mühlbacher-Karrer, M. Lucchi, M. Weyrer, L.-M. Faller, and H. Zangl, "Gesture-based contactless control of mobile manipulators using capacitive sensing," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE, 2020, pp. 1–6.

[95] H. Alagi, A. Heiligl, S. E. Navarro, T. Kroegerl, and B. Hein, "Material recognition using a capacitive proximity sensor with flexible spatial resolution," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 6284–6290.

[96] E. Zheng, J. Mai, Y. Liu, and Q. Wang, "Forearm motion recognition with noncontact capacitive sensing," *Frontiers in Neurorobotics*, vol. 12, p. 47, 2018.

[97] J. Cheng, O. Amft, G. Bahle, and P. Lukowicz, "Designing sensitive wearable capacitive sensors for activity recognition," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3935–3947, 2013.

[98] B. Chen, E. Zheng, X. Fan, T. Liang, Q. Wang, K. Wei, and L. Wang, "Locomotion mode classification using a wearable capacitive sensing system.," *IEEE Transactions on Neural Systems and Rehabilitation Engineering: a Publication of the IEEE Engineering in Medicine and Biology Society*, vol. 21, no. 5, pp. 744–755, 2013.

[99] X. Wang, Z. Liu, and T. Zhang, "Flexible sensing electronics for wearable/attachable health monitoring," *Small*, vol. 13, no. 25, p. 1 602 790, 2017.

[100] L. Viry, A. Levi, M. Totaro, A. Mondini, V. Mattoli, B. Mazzolai, and L. Beccai, "Flexible three-axial force sensor for soft and highly sensitive artificial touch," *Advanced Materials*, vol. 17, no. 26, pp. 2659–2664, 2014.

[101] A. Maslyczyk, J.-P. Roberge, V. Duchaine, *et al.*, "A highly sensitive multimodal capacitive tactile sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 407–412.

[102] P. Cataldi, S. Dussoni, L. Ceseracciu, M. Maggiali, L. Natale, G. Metta, A. Athanassiou, and I. S. Bayer, "Carbon nanofiber versus graphene-based stretchable capacitive touch sensors for artificial electronic skin," *Advanced Science*, vol. 5, no. 2, p. 1 700 587, 2018.

[103] J. Hashizume, T. M. Huh, S. A. Suresh, and M. R. Cutkosky, "Capacitive sensing for a gripper with gecko-inspired adhesive film," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 677–683, 2019.

[104] A. Gruebele, J.-P. Roberge, A. Zerbe, W. Ruotolo, T. M. Huh, and M. R. Cutkosky, "A stretchable capacitive sensory skin for exploring cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1750–1757, 2020.

[105] S. E. Navarro, S. Nagels, H. Alagi, L.-M. Faller, O. Goury, T. Morales-Bieze, H. Zangl, B. Hein, R. Ramakers, W. Deferme, *et al.*, "A model-based sensor fusion approach for force and shape estimation in soft robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5621–5628, 2020.

[106] H. Alagi, S. E. Navarro, M. Mende, and B. Hein, "A versatile and modular capacitive tactile proximity sensor," in *2016 IEEE Haptics Symposium (HAPTICS)*, IEEE, 2016, pp. 290–296.

[107] S. E. Navarro, S. Koch, and B. Hein, "3d contour following for a cylindrical end-effector using capacitive proximity sensors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 82–89.

[108] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[109] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 651–670, 1996.

[110] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[111] ——, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[112] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2010.

[113] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*, Springer, 2016, pp. 841–866.

[114] M. Azizian, M. Khoshnam, N. Najmaei, and R. V. Patel, "Visual servoing in medical robotics: A survey. part i: Endoscopic and direct vision imaging–techniques and applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 10, no. 3, pp. 263–274, 2014.

[115] M. Azizian, N. Najmaei, M. Khoshnam, and R. Patel, "Visual servoing in medical robotics: A survey. part ii: Tomographic imaging modalities–techniques and applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 11, no. 1, pp. 67–79, 2015.

[116] A. Krupa, J. Gangloff, C. Doignon, M. F. De Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE transactions on robotics and automation*, vol. 19, no. 5, pp. 842–853, 2003.

[117] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 1837–1842.

[118] B. Dahroug, B. Tamadazte, and N. Andreff, "Visual servoing controller for time-invariant 3d path following with remote centre of motion constraint," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 3612–3618.

[119] G. J. Garcia, J. Corrales, J. Pomares, F. Candelas, and F. Torres, "Visual servoing path tracking for safe human-robot interaction," in *2009 IEEE International Conference on Mechatronics*, IEEE, 2009, pp. 1–6.

[120] J. Pomares, J. A. Corrales, G. J. García, and F. Torres, "Direct visual servoing to track trajectories in human-robot cooperation," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, p. 44, 2011.

[121] Y. Wang, D. Ewert, R. Vossen, and S. Jeschke, "A visual servoing system for interactive human-robot object transfer," *Journal of Automation and Control Engineering Vol*, vol. 3, no. 4, 2015.

[122] T. Reichl, J. Gardiazabal, and N. Navab, "Electromagnetic servoing - a new tracking paradigm," *IEEE Transactions on Medical Imaging*, vol. 32, no. 8, pp. 1526–1535, 2013.

[123] N. F. Lepora, K. Aquilina, and L. Cramphorn, "Exploratory tactile servoing with active touch," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1156–1163, 2017.

[124] G. Sutanto, N. Ratliff, B. Sundaralingam, Y. Chebotar, Z. Su, A. Handa, and D. Fox, "Learning latent space dynamics for tactile servoing," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3622–3628.

[125] Y. Ding, F. Wilhelm, L. Faulhammer, and U. Thomas, "With proximity servoing towards safe human-robot-interaction," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 4907–4912.

[126] R. P. Joshi, N. Koganti, and T. Shibata, "A framework for robotic clothing assistance by imitation learning," *Advanced Robotics*, vol. 33, no. 22, pp. 1156–1174, 2019.

[127] A. Gabas, E. Corona, G. Alenyà, and C. Torras, "Robot-aided cloth classification using depth information and cnns," in *International Conference on Articulated Motion and Deformable Objects*, Springer, 2016, pp. 16–23.

[128] F. Zhang and Y. Demiris, "Learning grasping points for garment manipulation in robot-assisted dressing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 9114–9120.

[129] P. Jiménez and C. Torras, "Perception of cloth in assistive robotic manipulation tasks," *Natural Computing*, pp. 1–23, 2020.

[130] A. Zlatintsi, A. Dometios, N. Kardaris, I. Rodomagoulakis, P. Koutras, X. Papageorgiou, P. Maragos, C. S. Tzafestas, P. Vartholomeos, K. Hauer, *et al.*, "I-

support: A robotic platform of an assistive bathing robot for the elderly population," *Robotics and Autonomous Systems*, vol. 126, p. 103 451, 2020.

[131] L. Casas, C. Mürwald, F. Achilles, D. Mateus, D. Huber, N. Navab, and S. Demirci, "Human pose estimation from pressure sensor data," in *Bildverarbeitung für die Medizin 2018*, Springer, 2018, pp. 285–290.

[132] H. M. Clever, Z. Erickson, A. Kapusta, G. Turk, K. Liu, and C. C. Kemp, "Bodies at rest: 3d human pose and shape estimation from a pressure image using synthetic data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6215–6224.

[133] A. Kapusta, Z. Erickson, H. M. Clever, W. Yu, C. K. Liu, G. Turk, and C. C. Kemp, "Personalized collaborative plans for robot-assisted dressing via optimization and simulation," *Autonomous Robots*, pp. 1–25, 2019.

[134] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu, "Learning to collaborate from simulation for robot-assisted dressing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2746–2753, 2020.

[135] F. Achilles, A.-E. Ichim, H. Coskun, F. Tombari, S. Noachtar, and N. Navab, "Patient mocap: Human pose estimation under blanket occlusion for hospital monitoring applications," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 491–499.

[136] F. Zhang, A. Cully, and Y. Demiris, "Probabilistic real-time user posture tracking for personalized robot-assisted dressing," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 873–888, 2019.

[137] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 169–10 176.

[138] D. Göger, H. Alagi, and H. Wörn, "Tactile proximity sensors for robotic applications," in *2013 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, 2013, pp. 978–983.

[139] S. Mühlbacher-Karrer, A. Gaschler, and H. Zangl, "Responsive fingers—capacitive sensing during object manipulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 4394–4401.

[140] D. M. Taylor, "Americans with disabilities: 2014 (p70-152)," *U.S. Census Bureau*, 2014.

[141] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, A. E. Leeper, *et al.*, "Robots for humanity: Using assistive robotics to empower people with disabilities," *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, pp. 30–39, 2013.

[142] S. Schröer, I. Killmann, B. Frank, M. Völker, L. Fiederer, T. Ball, and W. Burgard, "An autonomous robotic assistant for drinking," in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 6482–6487.

[143] A. Kapusta, Z. Erickson, H. M. Clever, W. Yu, C. K. Liu, G. Turk, and C. C. Kemp, "Personalized collaborative plans for robot-assisted dressing via optimization and simulation," *Autonomous Robots*, 2019.

[144] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[145] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[146] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, *et al.*, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *arXiv preprint arXiv:1802.09464*, 2018.

[147] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," *http://pybullet. org*, 2016.

[148] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, "Dart: Dynamic animation and robotics toolkit.," *J. Open Source Software*, vol. 3, no. 22, p. 500, 2018.

[149] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 5026–5033.

[150] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[151] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *arXiv preprint arXiv:1903.11239*, 2019.

[152] F. Sadeghi, A. Toshev, E. Jang, and S. Levine, "Sim2real view invariant visual servoing by recurrent control," *arXiv preprint arXiv:1712.07642*, 2017.

[153] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 4243–4250.

[154] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei, "Surreal: Open-source reinforcement learning framework and robot manipulation benchmark," in *Conference on Robot Learning*, 2018, pp. 767–782.

[155] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied ai research," *arXiv preprint arXiv:1904.01201*, 2019.

[156] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[157] I. Zamora, N. G. Lopez, V. M. Vilches, and A. H. Cordero, "Extending the openai gym for robotics: A toolkit for reinforcement learning using ros and gazebo," *arXiv preprint arXiv:1608.05742*, 2016.

[158] A. Rodriguez-Ramos, C. Sampedro, H. Bavle, P. De La Puente, and P. Campoy, "A deep reinforcement learning strategy for uav autonomous landing on a moving platform," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1-2, pp. 351–366, 2019.

[159] D. Gallenberger, T. Bhattacharjee, Y. Kim, and S. S. Srinivasa, "Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2019, pp. 267–276.

[160] G. Canal, G. Alenyà, and C. Torras, "Personalization framework for adaptive robotic feeding assistance," in *International Conference on Social Robotics*, Springer, 2016, pp. 22–31.

[161] T. Rhodes and M. Veloso, "Robot-driven trajectory improvement for feeding tasks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2991–2996.

[162] I. Naotunna, C. J. Perera, C. Sandaruwan, R. Gopura, and T. D. Lalitharatne, "Meal assistance robots: A review on current status, challenges and future directions," in *2015 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, 2015, pp. 211–216.

[163] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[164] F. F. Goldau, T. K. Shastha, M. Kyrarini, and A. Gräser, "Autonomous multi-sensory robotic assistant for a drinking task," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, IEEE, 2019, pp. 210–216.

[165] P. M. Grice, A. Lee, H. Evans, and C. C. Kemp, "The wouse: A wearable wince detector to stop assistive robots," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2012, pp. 165–172.

[166] G. Canal, E. Pignat, G. Alenyà, S. Calinon, and C. Torras, "Joining high-level symbolic planning with low-level motion primitives in adaptive hri: Application to dressing assistance," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–9.

[167] N. Koganti, T. Tamei, K. Ikeda, and T. Shibata, "Bayesian nonparametric motor-skill representations for efficient learning of robotic clothing assistance," in *Workshop on Practical Bayesian Nonparametrics, Neural Information Processing Systems, 2016*, 2016, pp. 1–5.

[168] A. Clegg, "Modeling human and robot behavior during dressing tasks," Ph.D. dissertation, The Georgia Institute of Technology, 2019.

[169] A. Clegg, Z. Erickson, P. Grady, C. C. Kemp, G. Turk, and C. K. Liu, "Learning to collaborate from simulation for robot-assisted dressing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2746–2753, 2020.

[170] A. R. Tilley, *The measure of man and woman: human factors in design*. John Wiley & Sons, 2002.

[171] M. P. Lawton and E. M. Brody, "Assessment of older people: Self-maintaining and instrumental activities of daily living," *The gerontologist*, vol. 9, no. 3_Part_1, pp. 179–186, 1969.

[172] I. Akhter and M. J. Black, "Pose-conditioned joint angle limits for 3d human pose reconstruction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1446–1455.

[173] Y. Jiang and C. K. Liu, "Data-driven approach to simulating realistic human joint constraints," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1098–1103.

[174] A. Kapusta and C. C. Kemp, "Task-centric optimization of configurations for assistive robots," *Autonomous Robots*, pp. 1–22, 2019.

[175] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 72–83, 1987.

[176] J.-O. Kim and K. Khosla, "Dexterity measures for design and control of manipulators," in *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, IEEE, 1991, pp. 758–763.

[177] *Kinova jaco assistive robotic arm*.

[178] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, IEEE Press, 2013, pp. 301–308.

[179] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[180] E. A. Sisbot, L. F. Marin-Urias, X. Broquere, D. Sidobre, and R. Alami, "Synthesizing robot motions adapted to human presence," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 329–343, 2010.

[181] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, "Planning human-aware motions using a sampling-based costmap planner," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 5012–5017.

[182] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 1986–1993.

[183] G. Canal, G. Alenyà, and C. Torras, "Adapting robot task planning to user preferences: An assistive shoe dressing example," *Autonomous Robots*, vol. 43, no. 6, pp. 1343–1356, 2019.

[184] ——, "A taxonomy of preferences for physically assistive robots," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 2017, pp. 292–297.

[185] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A survey of preference-based reinforcement learning methods," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4945–4990, 2017.

[186] C. Wirth, J. Fürnkranz, and G. Neumann, "Model-free preference-based reinforcement learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[187] R. Akrour, M. Schoenauer, and M. Sebag, "Preference-based policy learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2011, pp. 12–27.

[188] T. Joppen, C. Wirth, and J. Fürnkranz, "Preference-based monte carlo tree search," in *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, Springer, 2018, pp. 327–340.

[189] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions.," in *Robotics: Science and Systems*, 2017.

[190] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017, pp. 4299–4307.

[191] R. Pinsler, R. Akrour, T. Osa, J. Peters, and G. Neumann, "Sample and feedback efficient hierarchical reinforcement learning from human preferences," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 596–601.

[192] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018.

[193] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[194] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[195] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 249–254.

[196] T. L. Chen, C.-H. A. King, A. L. Thomaz, and C. C. Kemp, "An investigation of responses to robot-initiated touch in a nursing context," *International Journal of Social Robotics*, vol. 6, no. 1, pp. 141–161, 2014.

[197] D. M. Taylor, "Americans with disabilities: 2014," *United States Census Bureau*, 2018.

[198] S. for Enhancing Geriatric Understanding, E. A. Surgical, and A. G. S. Medical Specialists (SEGUE), "Retooling for an aging america: Building the healthcare workforce," *Journal of the American Geriatrics Society*, vol. 59, no. 8, pp. 1537–1539, 2011.

[199] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[200] S. Schröer, I. Killmann, B. Frank, M. Völker, L. Fiederer, T. Ball, and W. Burgard, "An autonomous robotic assistant for drinking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 6482–6487.

[201] D. Park, Y. K. Kim, Z. M. Erickson, and C. C. Kemp, "Towards assistive feeding with a general-purpose mobile manipulator," *arXiv preprint arXiv:1605.07996*, 2016.

[202] C. J. Perera, T. D. Lalitharatne, and K. Kiguchi, "Eeg-controlled meal assistance robot with camera-based automatic mouth position tracking and mouth open detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1760–1765.

[203] T. Bhattacharjee, E. K. Gordon, R. Scalise, M. E. Cabrera, A. Caspi, M. Cakmak, and S. S. Srinivasa, "Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 181–190.

[204] R. P. Joshi, N. Koganti, and T. Shibata, "Robotic cloth manipulation for clothing assistance task using dynamic movement primitives," in *Proceedings of the Advances in Robotics*, ACM, 2017, p. 14.

[205] A. Jevtić, A. F. Valle, G. Alenyà, G. Chance, P. Caleb-Solly, S. Dogramadzi, and C. Torras, "Personalized robot assistant for support in dressing," *IEEE transactions on cognitive and developmental systems*, vol. 11, no. 3, pp. 363–374, 2018.

[206] R. M. Baños, C. Botella, M. Alcañiz, V. Liaño, B. Guerrero, and B. Rey, "Immersion and emotion: Their impact on the sense of presence," *Cyberpsychology & behavior*, vol. 7, no. 6, pp. 734–741, 2004.

[207] K. Kilteni, R. Groten, and M. Slater, "The sense of embodiment in virtual reality," *Presence: Teleoperators and Virtual Environments*, vol. 21, no. 4, pp. 373–387, 2012.

[208] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 5495–5502.

[209] J. Wainer, D. J. Feil-Seifer, D. A. Shell, and M. J. Mataric, "The role of physical embodiment in human-robot interaction," in *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2006, pp. 117–122.

[210] V. Kumar and E. Todorov, "Mujoco haptix: A virtual reality system for hand manipulation," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2015, pp. 657–663.

[211] J. I. Lipton, A. J. Fay, and D. Rus, "Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 179–186, 2017.

[212] V. Kumar, A. Gupta, E. Todorov, and S. Levine, "Learning dexterous manipulation policies from experience and imitation," *arXiv preprint arXiv:1611.05095*, 2016.

[213] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018.

[214] H. Liu, Z. Zhang, X. Xie, Y. Zhu, Y. Liu, Y. Wang, and S.-C. Zhu, "High-fidelity grasping in virtual reality using a glove-based system," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5180–5186.

[215] A. Frisoli, M. Bergamasco, M. C. Carboncini, and B. Rossi, "Robotic assisted rehabilitation in virtual reality with the l-exos," *Stud Health Technol Inform*, vol. 145, pp. 40–54, 2009.

[216] K. Brütsch, A. Koenig, L. Zimmerli, S. Mérillat-Koeneke, R. Riener, L. Jäncke, H. J. van Hedel, and A. Meyer-Heim, "Virtual reality for enhancement of robot-assisted gait training in children with neurological gait disorders," *Journal of rehabilitation medicine*, 2011.

[217] B. R. Ballester, L. S. Oliva, A. Duff, and P. Verschure, "Accelerating motor adaptation by virtual reality based modulation of error memories," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, IEEE, 2015, pp. 623–629.

[218] X. Huang, F. Naghdy, G. Naghdy, H. Du, and C. Todd, "The combined effects of adaptive control and virtual reality on robot-assisted fine hand motion rehabilitation

in chronic stroke patients: A case study," *Journal of Stroke and Cerebrovascular Diseases*, 2018.

[219] F. Bernardoni, Ö. Özen, K. Buetler, and L. Marchal-Crespo, "Virtual reality environments and haptic strategies to enhance implicit learning and motivation in robot-assisted training," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, IEEE, 2019.

[220] D. Park, Y. Hoshi, H. P. Mahajan, H. K. Kim, Z. Erickson, W. A. Rogers, and C. C. Kemp, "Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned," *Robotics and Autonomous Systems*, vol. 124, p. 103 344, 2020.

[221] C. K. Liu and D. Negrut, "The role of physics-based simulators in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2020.

[222] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, Jul. 2020.

[223] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 975–10 985.

[224] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5253–5263.

[225] S. Liu, Y. Yin, and S. Ostadabbas, "In-bed pose estimation: Deep learning with shallow dataset," *IEEE journal of translational engineering in health and medicine*, vol. 7, pp. 1–12, 2019.

[226] Z. Erickson, S. Chernova, and C. C. Kemp, "Semi-supervised haptic material recognition for robots using generative adversarial networks," in *Conference on Robot Learning*, PMLR, 2017, pp. 157–166.

[227] Z. Erickson, N. Luskey, S. Chernova, and C. C. Kemp, "Classification of household materials via spectroscopy," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 700–707, 2019.

[228] Z. Erickson, E. Xing, B. Srirangam, S. Chernova, and C. C. Kemp, "Multimodal material classification for robots using spectroscopy and high resolution texture imaging," *arXiv preprint arXiv:2004.01160*, 2020.